RE
Transactions
ELECTRONIC COMPUTERS

Volume EC-9 JUNE, 1960 Number 2

Published Quarterly

TABLE OF CONTENTS

CONTRIBUTIONS

CORRESPONDENCE

PUBLISHED BY THE
Professional Group on ELECTRONIC COMPUTERS

IRE Transactions® on Electronic Computers

Published by the Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79th Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Price per copy: IRE-PGEC members, $1.80; IRE members $2.70, nonmembers, $5.40. Yearly subscriptions rate: nonmembers, $17.00; colleges and public libraries, $12.75. Address requests to The Institute of Radio Engineers, 1 East 79th Street, N.Y. 21, N.Y.

Notice to Authors: Address all papers and editorial correspondence to H. E. Tompkins at the address on the inside back cover. To avoid delay and inconvenience in the processing of manuscripts, please follow the procedure suggested there.

Sequence Detection Using All-Magnetic Circuits*

H. D. CRANE†, SENIOR MEMBER, IRE

Summary—A technique is described for detecting specific sequences of pulses occurring on a net of input lines. This technique lends itself to realization in all-magnetic networks by the use of multi-aperture magnetic devices (MAD's). The resulting circuits are remarkably simple and reliable. Processing rates in excess of 10,000 characters per second may be achieved. Examples are given of systems using arrays of such detectors. One example involves a system for detecting handwritten characters which makes use of a special pen having the property of generating specific sequences of pulses as symbols are written. The second example relates to the problem of monitoring text for the detection of specific words (letter sequences) and phrases (series of sequences).

Introduction

A LOGIC system developed at SRI[1] utilizing multi-aperture magnetic devices (termed MAD's) can be applied with advantage to the problem of detecting specific sequences of pulses on a network of input lines.

The operational and design features of these circuits have been covered in other publications.[1,2] One of their most important features is that binary data transfer between elements is achieved over coupling loops containing only conductive wire. In this system, a shift register takes the form, simply indicated in Fig. 1, where alternate elements are labelled O (odd) and E (even), and the driver consists of four clocks: two Advances (Adv $O \rightarrow E$ and Adv $E \rightarrow O$) and two Clears (Clear O and Clear E). The transfer process is nondestructive in the transmitter so that the transmitting element must be explicitly cleared before being read into. Thus the basic clock cycle is

$$\cdots \text{Adv } E \rightarrow O, \text{ Clear } E, \text{ Adv } O \rightarrow E, \text{ Clear } O, \cdots.$$

Type 1 Sequence Detector

The aim is the detection of a precise (previously specified) sequence of pulses, should this sequence of pulses ever appear on the network of input lines. The detection network, Fig. 2(a), is structurally similar to the (shift register) chain of Fig. 1. The correct sequence of input pulses results in the propagation of a binary *one* completely down the chain. For simplicity, let the input lines be labeled by letters of the alphabet. Then the network of Fig. 2(a) will detect the sequence AUTOMAT. A current on the A line will enter a binary *one* into the beginning of the structure. A current on the U line will advance the *one* to the second stage. Successive currents on the T, O, M, A, T lines advance the *one* to the last stage. A suitable indicator connected to the last stage indicates detection of the sequence wired into the structure. However, this network will also respond whenever a sequence consisting of AUTOMAT preceded by, interspersed with, or followed by, other letters is received. For example, AUTO*B*MAT would be detected since currents on unused letter lines (B, C, D, $\cdots$) have no effect on the network. Similarly AU*A*TOMAT would be detected since extra occurrences of used letters (A, M, O, $\cdots$) leave the state of the network unchanged.

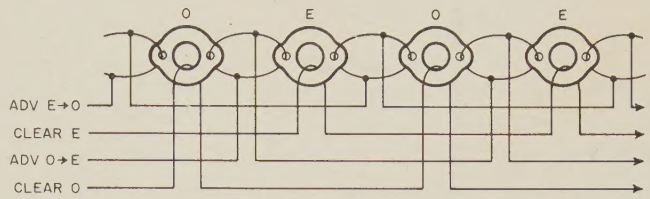
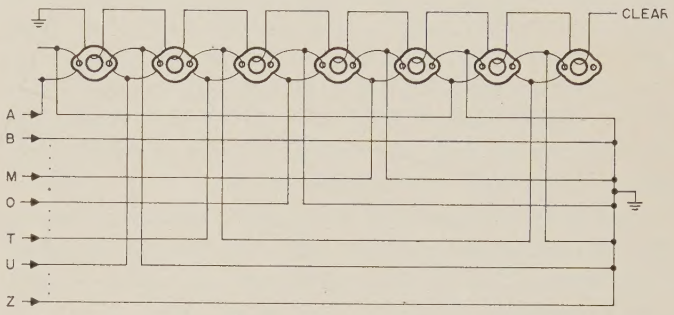
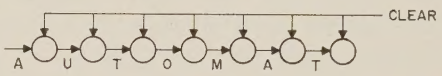


Fig. 1—MAD shift register.



(a)



(b)

Fig. 2—Type 1 sequence detector.

Thus a Type 1 detector is one that detects a specific sequence of pulses interspersed by any other pattern of pulses. Note that as each character of the desired sequence is received, the *one* state advances a single position, but the previous elements remain in the *one* state. A single Clear winding chain is provided for clearing the entire detector at once.

A schematic form for the Type 1 detector is illustrated in Fig. 2(b), where the MAD's are represented by

* Received by the PGEC, September 22, 1959. The work reported here was carried on at Stanford Res. Inst. under the sponsorship of the Systems Components Division of AMP Inc., Harrisburg, Pa.

† Stanford Res. Inst., Menlo Park, Calif.

[1] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," PROC. IRE, vol. 47, pp. 63–73; January, 1959.

[2] D. R. Bennion and H. D. Crane, "Design and analysis of MAD transfer circuitry," *Proc. WJCC*, pp. 21–36; March, 1959.

circles and the coupling loops by arrowed lines. The input line that excites any particular coupling loop is indicated by the letter symbol adjacent to the loop.

Type 2 Sequence Detector

The network of Fig. 3(a) is similar to that of Fig. 2(a) except that the Clear winding is so connected as to be excited by every input pulse. Therefore at every current pulse there is a tendency to clear out the entire network. Suppose that the sequence AUT has already been received so that the third element is in the *one* state. An *O* current must override the Clear current and advance the *one* to the fourth element. This is simply done by having the *O* current flow through windings opposing the Clear mmf in the third and fourth elements; in this way the transfer may be successfully achieved. However, if any letter other than *O* were received after the partial sequence AUT, then the network would be completely cleared to *zero*. Each stage is similarly connected. By this means, then, the correct sequence, and only the correct sequence, will register success.

A Type 2 detector will thus detect the occurrence of the desired sequence, but *only* the desired sequence. Note that a Type 2 detector is somewhat more complicated to construct than a Type 1 detector.

The schematic arrangement of a Type 2 detector, Fig. 3(b), is similar to that for a Type 1, except that the source of Clear current in Type 2 detectors is indicated as coming from the input lines themselves.

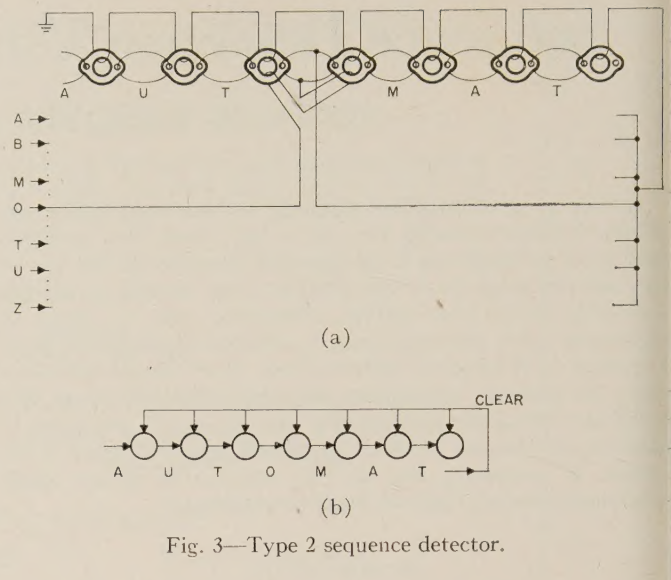


Fig. 3—Type 2 sequence detector.

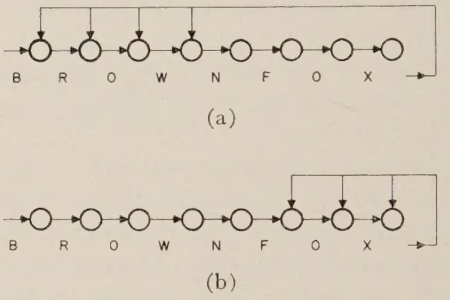


Fig. 4—Type 3 sequence detector.

Type 3 Sequence Detector

Type 1 and Type 2 detector techniques may be used together in a single detector. For example, the detector of Fig. 4(a) will indicate success only if the word BROWN is detected first, exactly as shown, followed by detection of FOX, possibly interspersed by other characters. In Fig. 4(b), the network is reversed so that success is indicated only if the sequence BROWN is first detected, with other characters possibly interspersed, followed by FOX exactly as shown. For simplicity, no provisions for clearing the entire detector are indicated.

Mixed detectors of this type can be realized with an arbitrary number of alternating Type 1 and Type 2 sequences in series.

Basic Sequence Detector Properties

Before extensions and applications of this detection approach are considered, some of the basic detector properties are indicated.

The detector circuits are substantially unaffected by the number of pulses appearing in sequence on any input line, provided that no pulses occur on other lines during the same interval. Any single pulse completely performs its function of advancing, or clearing, or both. Another pulse occurring on the same line, before another pulse appears on a different line, has no effect on the circuit. Thus the sequence AAUUUTOMMATT would registe success by either type of detector wired for AUTOMAT It is interesting to note as a corollary of this fact tha the detector circuits are "bounce" insensitive, if th sources of input currents are, in fact, from mechanica switches.

The MAD transfer system does not permit simultane ous read-in and read-out from a given element (in fact the success of the transfer mode is based upon the fac that the output is effiectively decoupled during entry) Therefore, the same input line cannot be connected t both sides of a given element. Thus, without furthe consideration, it would not be possible to design detec tors for sequences in which a given character repeats i adjacent positions. Two techniques are described fo handling this situation.

In the arrangement of Fig. 5(a), an extra input line *P*, is provided. This line is excited after every inpu pulse, but before the next input pulse. Thus an in put sequence 27774433332 results in the pulse sequenc 2P7P7P7P4P4P3P3P3P3P2P to the detector. However the detector need provide for *P* elements between repea characters only. Thus the detector has the form 27P7P74P43P3P3P32. This technique is applicable t both Type 1 and Type 2 detectors. For a Type 2 de tector, Fig. 5(a), the clear line is excited by all inpu

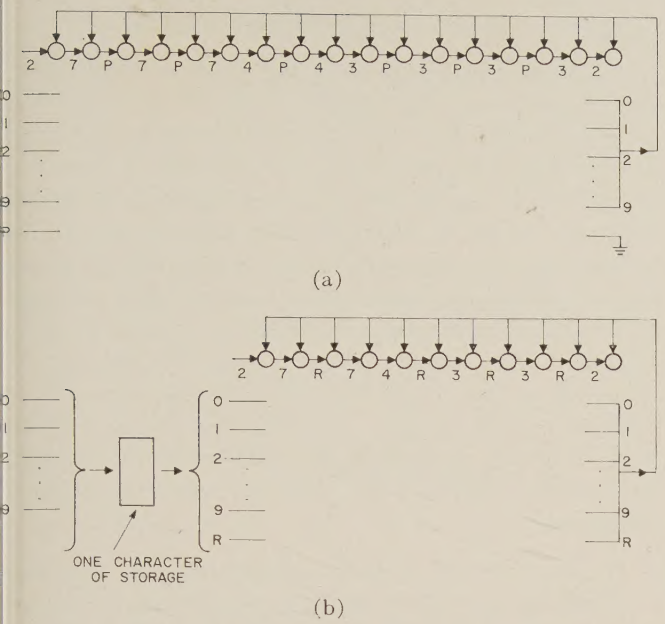


Fig. 5—Sequences with repeated characters. Illustrated for sequence 27774433332.

ines, except the *P* line. If the *P* line were included in the learing, then *P* elements would be required after *every* haracter (element) of the sequence. This technique reluces the maximum allowed input rate by one-half.

In the second arrangement, Fig. 5(b), an extra input ne, *R*, is provided. Means must be provided for generting an *R* (repeat) signal to replace the second (fourth, ixth, etc.) pulse of a repeated sequence. One character of storage is required, so that if the present character grees with the last character, then a Repeat signal is ssued instead. A Repeat signal clears the storage so that he following signal is handled unchanged. Only a single Repeat line is required, regardless of the number of input ines. The same example of a repeated sequence is illusrated in Fig. 5(b). The input sequence 27774433332 esults in the pulse sequence 27R74R3R3R2 to the dector. With this approach, there is no reduction in the naximum character rate, nor are extra elements required; however, it is not applicable to Type 1 detection.

There are alternative ways of indicating successful letection of a sequence. If sequence detection is only he first step of a processing system, then electronic dection of success is very simple with the detected signal eing fed into other networks for further processing. If equence detection is itself the end product, suitable ndicators may be connected to the last element of the letector chain. The nondestructive read-out possibiliies of MAD's permits excitation of a number of devices equiring considerable power, such as incandescent ulbs.[1,3]

Detectors of the type thus far described can be used:) in systems in which the input consists of an arbitrary equence of known sequences; or 2) in systems in which there is no *a priori* knowledge of the content of the input. Examples of both types of systems are illustrated below, by means of which still other detector properties are derived.

When MAD elements constructed from typical ferrite memory materials are used, input pulses of 1- to 5-μsec duration are adequate for proper operation, depending upon the specific material. Thus detectors in this form are capable of handling character rates in excess of 100,000 per second. For MAD elements of 0.2 inch average diameter, using memory-type ferrites that switch in 2–5 μsec, input (or setting) mmf on the order of one ampere-turn is suitable. Good clearing of elements requires two or three times this magnitude of mmf.

Handwriting Detector System

To illustrate the first of the system types, an experimental real-time handwriting detection scheme[4] is described which uses a direction-sensitive pen, Fig. 6. This particular pen is sensitive to five directions of motion: N, S, E, W, V, where N, S, E, W indicate the common geographical directions, and V indicates up-down motion. As symbols are printed in normal fashion, the pen signals the present direction of motion by having a commutator close against a contact in that direction of writing. Each contact covers approximately 90 degrees: from northeast to southeast, from southeast to southwest, etc. For each written symbol, a particular sequence of pulses is generated on the five output lines. Equivalently, the pulse source appears as a single-pole five-position switch, only one position of which may be closed at a time, as in Fig. 7(a). Then as many independent detectors as necessary may be connected to monitor the emitted sequences simultaneously. In Fig. 7(a), ten Type 1 detectors are indicated, labeled 0 through 9, to detect in real-time the writing of any digit. The simpler Type 1 detectors are adequate here because of the small number of known sequences permitted at the input. The particular sequences being searched for are indicated in Fig. 7(b), although other detectors could be used to search simultaneously for a large number of other symbols as well. For demonstration purposes, neon bulbs are used for the detector indicators in the detector array shown photographed in Fig. 6. This array searches for the ten sequences, 0 through 9. The neon bulb at the output of each detector is biased by the detector power source voltage [of Fig. 7(a)] which is chosen to be sufficient to sustain breakdown, *e.g.*, 67½-volt battery. In this particular sequence array, the up-down sensor is not utilized. Furthermore, the detectors are shortened in length to the portion to the right of the dashed line in Fig. 7(b). If the writer limits his vocabulary to only the digits 0 through 9, the portions to the left of the line are not

[3] J. A. Rajchman and A. W. Lo, "The Transfluxor," Proc. IRE, ol. 44, pp. 321–332; March, 1956.

[4] This experimental handwriting detection scheme was developed at SRI, under SRI sponsorship.

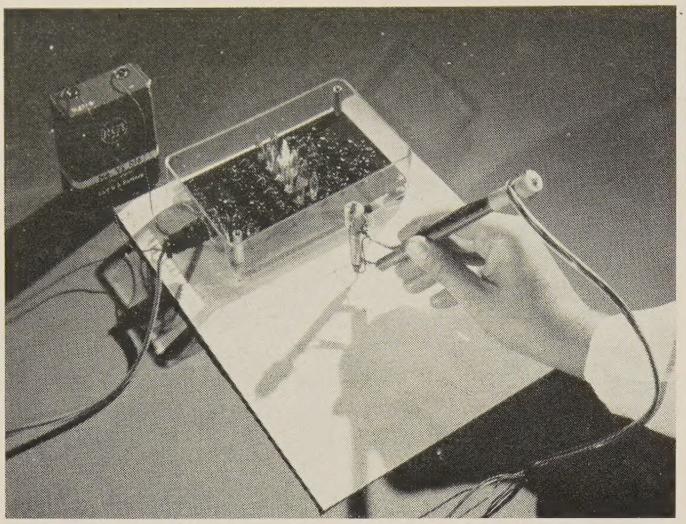
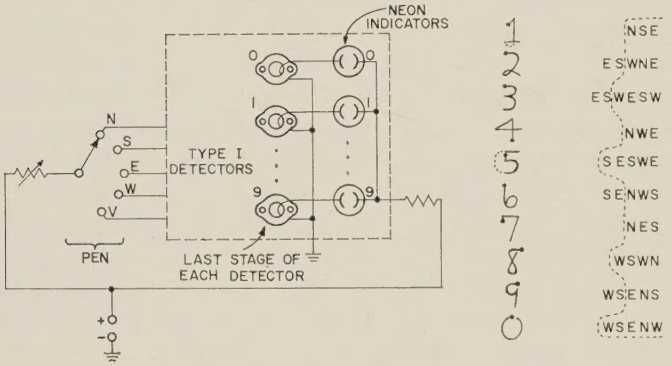


Fig. 6—Photograph of handwriting detection unit.



(a) (b)

Fig. 7—Handwriting detection scheme.

necessary for positive identification of each of the ten symbols. Thus the abridged list of ten sequences shows 36 letter (or direction) occurrences, implying the use of 36 magnetic elements distributed in the ten detectors. However, in this particular arrangement, use of neon indicators makes it possible to save one element per detector, accounting for the total of 26 elements in the photograph. The bistable neon tube is connected in such a way as to serve effectively as the last element of each sequence chain. This reduction is possible because of the biased neon indicator circuits. The winding polarity is arranged so that *one* transfer into the last element causes a voltage in the neon circuit opposing the voltage source bias. The final character line is connected to clear the last element, the resulting signal of which aids the bias voltage in the indicator circuit and therefore results in breakdown of the indicator.

The ten Type 1 detectors may be cleared simultaneously after any digit is detected, and the detectors are then ready to search for the next digit.

Further reduction in the number of elements can be achieved by use of the technique discussed below in connection with Fig. 9.

Message Monitoring System

A system characteristic of the type in which there is no *a priori* knowledge of the content of the input is one concerned with the real-time monitoring of text for detecting the occurrence of various kinds of sequences. Consider the continuous character sequence to be specified only to the extent that it is broken up into individual messages which are separated by end-of-message (EOM) symbols. Furthermore, suppose each message is broken up into chunks of characters (words) separated by space (Sp) signals. Then one may be interested in detecting various types of words and sequences of words (phrases). The number of word and phrase sequence types is large, and there is no particular advantage in generating a long list here. Rather, just a few types are mentioned.

1) Detection of a particular sequence of characters [either a) within a word, or b) within the entire message], regardless of what other characters are interspersed. This may be of interest for code detection.
2) Detection of a root word regardless of suffix or prefix; for example, detection of the root word AUTOMAT from the occurrence of any of the following words (preferably by use of a single detector):

AUTOMAT
AUTOMATIC
AUTOMATE
NONAUTOMATIC
AUTOMATION
etc.

3) Detection of a given sequence only if it appears between successive Sp marks exactly as specified, *e.g.*, to detect SpAUTOMATESp.
4) Detection of a sequence of words such as NEW YORK CITY.

All of these word and phrase sequence types can be realized by appropriate combinations of Type 1, 2, and 3 detectors, suitably interconnected. The array of detectors then constitutes a dictionary, each word and phrase of which is searched for during each message. Thus the search process may be to clear all detectors at each EOM symbol, and to search for and record all detected sequences during the following message. The sequences detected during each message can be stored and recorded with the message in various forms, or can be used for further message routing and processing.

Realizations of the sequences indicated above are illustrated in Fig. 8. Although it is not specifically shown, all detectors are cleared at EOM. The detectors have other Clear signals as well during the message, the Clear logic being determined by the kind of sequence being searched for. For example, the detector of Fig

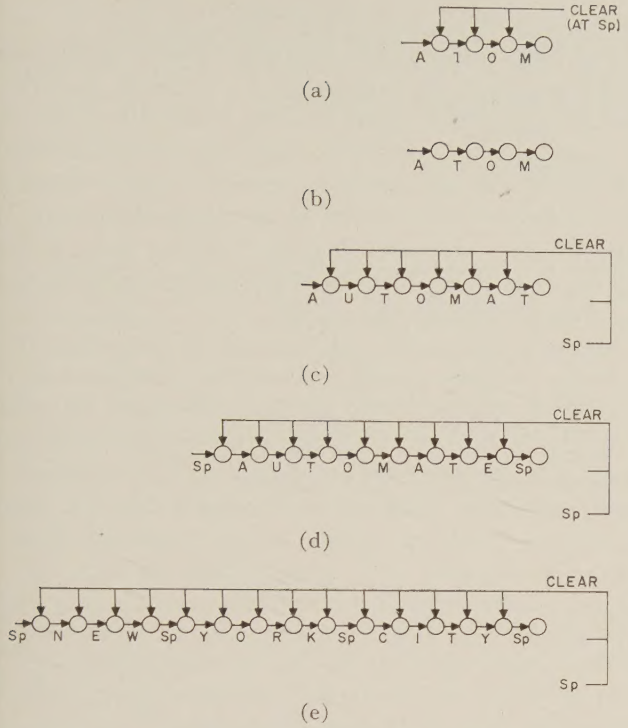


Fig. 8—Realizations of various sequence types.

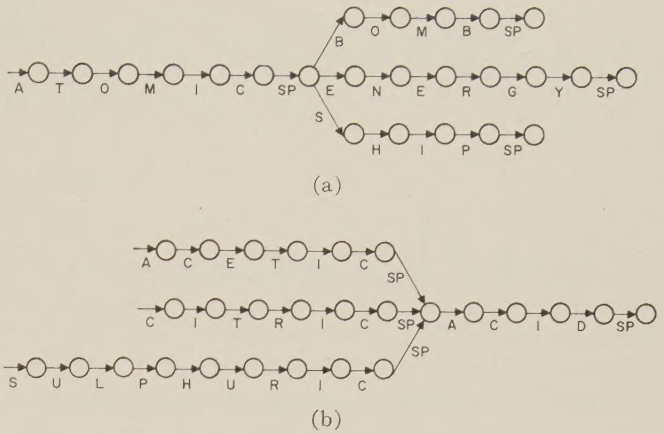


Fig. 9—Fan-in, fan-out arrangements.

8(b) has no other Clear but at EOM. In particular, the last element of each detector is not cleared until EOM. Thus, if success is ever achieved, the fact is stored until EOM. In some detectors, however, it may be necessary to clear even the last stage at every Sp signal, if, for example, it is of interest to count the number of occurrences of the same word in the message. Other types of phrase structure may also be of interest. For example, it may be desirable to arrange the clearing so that extraneous words may appear between some (or all) of the specified words of the phrase.

Note that the Sp character is treated the same as any other character. For example, in the sequence of Fig. 8(d) the Sp signal is actually the first character of the detector. Furthermore, in the Type 2 detectors of Fig. 8(c)-(e) the Sp line is shown (schematically) connected to the general Clear line.

The fan-in and fan-out facility of MAD logic can afford some economy in phrase detection where

1) the same word modifies a number of other words, Fig. 9(a).
2) a given word is acceptable only if modified by a selected list of words, Fig. 9(b).

Both of these situations are illustrated in Fig. 9. Use of separate apertures for the fan-in and fan-out windings results in no interaction between the various windings.

A simple data-retrieval system, based on uniterms (descriptors), serves as an example of a particular system meeting the specifications listed above for a message monitoring system. A list of abstracts (messages) is serially listed on a storage medium such as a magnetic tape. Also provided are wired detectors for all possible uniterms of interest. The user then assembles the appropriate detectors to search for his particular uniterms of interest, which may number a dozen or so. If the uniterms of interest are designated as $x_1, x_2, \cdots, x_n$, then there may be interest in a particular abstract only if the uniterms detected satisfy some specified logic connective as $(x_1+x_2)\bar{x}_3+x_4+\cdots$. In this case, the detector success signals are processed in a straightforward manner at EOM, so that the particular abstract is flagged only if the logic expression is satisfied. Because of the general logic facility of MAD elements,[1] the logic connective itself can also be generated in all-magnetic circuits.

One feature of such an approach is that the character lines excited from the tape can be simultaneously and independently used by any number of other searchers. Each searcher can "hook on" at any time. He need only remain connected for as long as it takes to scan completely the list of abstracts recorded on the tape.

Error Neglect

An important feature of detector systems is their ability to handle (or neglect) errors. Where the input lines are excited from equipment having certain probabilities of error (either human or equipment), then it may be desirable to be able to detect key words even with a single error or more in the sequence.

By making use of the fan-in, fan-out properties of the MAD elements, as well as the techniques preiviously discussed for handling repeat characters, error-neglect detectors are easily constructed. The character repeat scheme of Fig. 5(a) is used in forming the single-error-neglect detector of Fig. 10(a), where the detector searches for the word MISSILE. The elements shown shaded in the figure are provided only to prevent the same letter from connecting to the input and output of the same element. The extra internal pulse generated after every input pulse is labeled P.

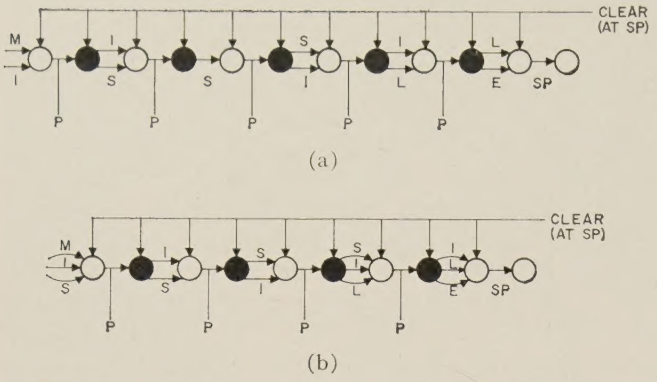


Fig. 10—(a) Single and (b) double error-neglect detectors.

It is clear that the detector of Fig. 10(a) will detect the word MISSILE with any one single error. An *M* or *I* will introduce a *one* into the first stage. The automatic *P* pulse following the *M* or *I* will advance the *one* to the second stage. An *I* or *S* will advance the *one* to the third stage, etc. Thus any of the sequences

⊗ISSILE
M⊗SSILE
MI⊗SILE
MIS⊗ILE
MISS⊗LE
MISSI⊗E
MISSIL⊗

will register success, where the symbol ⊗ stands for *any* letter.

By arranging for fan-in and fan-out of 3, double-error-neglect detectors are realized, as indicated in Fig. 10(b).

Conclusion

The logic approach to sequence detection discussed in this paper can be realized by a number of alternative circuit approaches. However, the reduction to practice by multi-aperture magnetic elements (MAD's) leads to remarkable circuit simplicity, as well as great reliability. The former is primarily due to the combined storage and logic facility of these elements. It has been demonstrated that these detectors can handle character rates in excess of 100,000 per second.

Where a large array of detectors is required, as for example in some message monitoring systems, this approach has the disadvantage that the search dictionary is not easily altered. It has the positive advantage, however, of providing high character rates because of the directness of its approach; no general data-processing techniques requiring random-access memory facility as well as arithmetic and programming facility are required. Furthermore, arrays of this type are very easily constructed and may be packaged into small volume, so that search dictionaries may be easily changed.

MAD's can be used for performing general logic.[1] This allows for a great deal of flexibility in interconnecting sequence detectors. For example, as discussed in the text, the OR input property of MAD's allows for the design of built-in error-neglect facilities which may be very powerful in certain applications.

Acknowledgment

The author wishes to acknowledge the significant contributions of his colleague, Dr. David R. Bennion, to this work. The permission of AMP Incorporated for release of the material presented here is also gratefully acknowledged.

Comparison of Saturated and Nonsaturated Switching Circuit Techniques*

G. H. GOLDSTICK†, MEMBER, IRE

Summary—The concept that the junction transistor is a charge-controlled current source is reviewed. Saturated operation and nonsaturated operation are defined on the basis of minority and majority carrier distributions in the base region. Several common emitter switching circuits are analyzed. The switching efficiency, a figure of merit based on the charge storage properties of the transistor, is introduced. Saturated and nonsaturated operation are compared on the basis of switching efficiency, transient waveforms, stability of the voltage levels, power dissipation, noise rejection and suppression ability, and circuit complexity. Currently-used antisaturation techniques are discussed.

Introduction

EVER since transistors were first used as switches in computer circuits, a controversy has existed as to the relative merits of saturated vs nonsaturated operation. Unfortunately, the arguments raised on both sides are often qualitative, and as a result certain misconceptions and misunderstandings as to the merits of the two techniques have developed. This has resulted in proponents of nonsaturated operation dismissing saturated circuits with the comment "they're too slow," and proponents of saturated operation dismissing nonsaturated circuits with the comment "they're unreliable." The relative merits of the two techniques, of course, depend upon the specific application.

It is usually recognized that saturated circuits are capable of lower power dissipation and greater voltage stability than the nonsaturated type. They are also lower in cost and simpler in design. Saturated circuits are, however, inferior in speed because of greater minority carrier storage which results in slower turn-off times. Greater switch-off speed (or, as will be shown below, switching efficiency) is thus the one significant advantage which nonsaturated circuitry has over saturated circuitry. It will be a primary purpose of this paper to investigate how real this advantage is and under what conditions it exists. An attempt will also be made to clarify and assess the issues, and to provide some engineering insight and quantitative results which will aid the designer in evaluating the two techniques under the conditions posed by his particular problem.

In order to accomplish this objective, concepts relating to the transistor as a charge-controlled device will be thoroughly reviewed in order to compare quantitatively the merits of both types of circuitry. In accomplishing this comparison, it will be convenient to introduce the switching efficiency E_s, a figure of merit of the transistor switch which will be based on the charge properties of the semiconductor. The switching efficiency will prove to be quite valuable in comparing the saturated and nonsaturated techniques. Some of the common antisaturation techniques which have been introduced over the past several years will then be reviewed to point out the various design requirements and performance limitations on these circuits. Distortion, voltage stability, and power dissipation properties of the common antisaturation feedback clamp will also be investigated.

The detailed discussion in this paper will be restricted to the common emitter switch which provides current gain, voltage gain and load isolation. Circuit techniques which employ the transistor in the common collector and common base configurations [16] will not be subject to detailed examination; however, the computations and criteria which will be developed are certainly applicable to these circuits.

Charge-Controlled Concepts

The basic concepts relating to the transistor as a charge-controlled current source have been previously established [1]–[6]. These principles are summarized in the following paragraphs.

The transistor's three large signal regions have been defined for the junction transistor as follows [7], [8]:

1) Region I: (Cut-off) emitter and collector junction reverse biased.
2) Region II: (Active) emitter junction forward biased, collector junction reverse biased.
3) Region III: (Saturated region) emitter and collector junctions both forward biased.

Fig. 1 illustrates the minority carrier densities associated

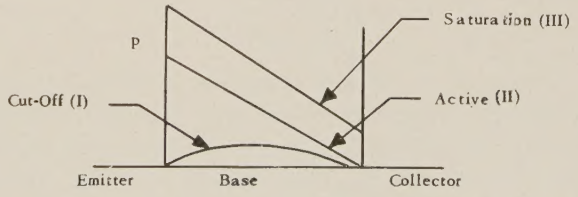


Fig. 1—Carrier densities in the base region corresponding to three operating regions.

with each of the three regions of operation. The conditions on the minority carrier densities at the junctions for a *p-n-p* transistor may be summarized as follows:

* Received by the PGEC, December 2, 1959; revised manuscript received, March 3, 1960. This paper was originally presented at the 1959 Northeast Electronics Research and Engineering Meeting, Boston, Mass., November 19, 1959.

† Electronics Div., The National Cash Register Co., Hawthorne, Calif.

1) Cut-off region: $p(x=0) \ll pn$, $p(W) \ll pn$.
2) Active region: $p(0) \gg pn$, $p(W) \ll pn$.
3) Saturated region: $p(0) \gg pn$, $p(W) \gg pn$.

Saturation is most often defined as the condition of the transistor when the density of holes at the collector junction exceeds the equilibrium density; *i.e.*, when the actual internal collector junction becomes forward biased so that the collector-to-emitter drop is less than the base-to-emitter drop. This definition is quite adequate under nominal dc circuit conditions, but is much too narrow when transient and recovery problems are considered. Instead, we will refer to a transistor as saturated when the control charge in the base region exceeds that which is required to support the terminal dc collector current. This definition is identical to the previous one under steady-state conditions, but differs significantly under transient conditions. This definition of saturation will prove to be of value in comparing saturated and nonsaturated techniques.

When a transistor is operated in the active region, there is a one-to-one correspondence between the collector current and the charge stored in the base region. Under the assumptions that the emitter efficiency is unity, the transport factor is approximately unity, and the base width is small in comparison to the diffusion length

$$\left(\frac{W}{L_p} \ll 1\right),$$

then the base charge is: [1], [4], [9][1]

$$Q = \tau I_e \tag{1}$$

where τ is the base transit time (the average time spent per carrier in the base region), and I_e is the steady-state emitter current. In this paper, we will restrict ourselves to uniformly-doped junction transistors, whose base transit time τ is given by

$$\tau = \frac{1.22}{\omega_{\alpha_0}},$$

where ω_{α_0} is the grounded base, short-circuit current gain cut-off frequency measured at $V_{cb}=0$ volt.

Since $I_c=\alpha_0 I_e$, the base control charge is

$$Q = \frac{1.22 I_c}{\omega_{\alpha_0}\alpha_0}. \tag{2}$$

The process of "turning a transistor on" (*i.e.*, allowing it to become capable of conducting a given dc current, I_c, in the active region) consists of inserting into the base region the charge $(1.22I_c)/(\omega_{\alpha_0}\alpha_0)$; whereas that of "turning a transistor off" consists of removing the charge $(1.22I_c)/(\omega_{\alpha_0}\alpha_0)$. The speed with which a transistor may be made capable of conducting a given current, I_c, in the active region and then be cut off is theoretically limited [for times $t>(2.43/\omega_\alpha)$] only by the speed with which the external circuitry is capable of supplying and removing the required charge.

When a transistor is operated in the saturation region, the control charge which is stored in the base region for a given collector current, I_c, is increased above that which existed in the active region by an amount which depends upon the excess base current, I_{bx}, defined as I_b-I_c/β. The excess base current results in the base collector diode becoming forward biased, and therefore in the total collector-to-emitter voltage V_{ce} being decreased. These phenomena are demonstrated in the collector characteristics of Fig. 2 in which it is observed that increasing the base current at a constant collector current results in a decreasing collector-to-emitter voltage. In saturation, therefore, the collector-to-emitter voltage may be expressed as a function of I_c and I_b [7]:

$$V_{ce} = V_{ce}(I_c, I_b). \tag{3}$$

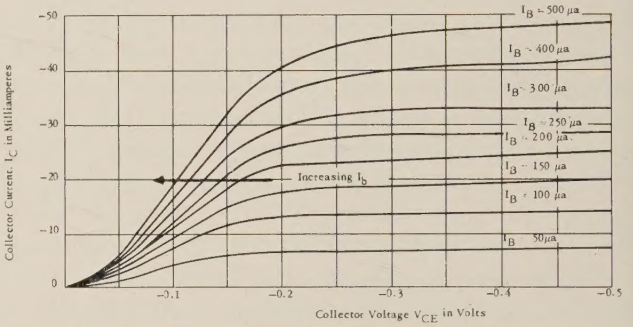


Fig. 2—Saturation region collector characteristics.

It is shown in Appendix I that the control charge under steady-state conditions may be expressed as follows:

$$Q_b = \frac{1.22 I_c}{\alpha_0\omega_{\alpha_0}} + K_s\left[I_b - \frac{I_c}{\beta_0}\right], \tag{4}$$

where

I_c is the dc collector current,
I_b is the dc base current,
β_0 is the short-circuit common emitter current gain,
α_0 is the short-circuit common base current gain,
ω_{α_0} is the grounded base current gain cut-off frequency measured at about zero volt, and
K_s is the saturation time constant; *i.e.*, the average time for holes to diffuse to the junction or recombine when the transistor is saturated.

It can be observed that Q_b is a linear function of both I_c and I_b.

Eq. (4) describes the charge stored in the base region under steady-state conditions; it is shown in Appendix I that if a charge

$$Q_0 > \frac{1.22 I_c}{\alpha_0\omega_{\alpha_0}}$$

[1] See Appendix I. (For a more general derivation, see Beaufoy and Sparkes [1].)

exists in the base at time $t=0$, the charge which will be stored in the base at time t is

$$Q_b(t) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left[Q_0 - \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}}\right]e^{-t/K_s} + K_s\left[I_b - \frac{I_c}{\beta}\right][1 - e^{-t/K_s}]. \quad (5a)$$

As $t\to\infty$,

$$Q(\infty) = \frac{1.22I_c}{\omega_{\alpha_0}\alpha_0} + K_s\left[I_b - \frac{I_c}{\beta}\right], \quad (5b)$$

which is in agreement with (4). It is observed from (5) that the excess charge stored in the base decays with the time constant t/K_s. Eq. (5b) has been verified experimentally for a series of uniformly-doped alloy junction transistors [4].

The principal factors which limit the speed with which a transistor may be saturated to a given voltage and current and then cut off may be summarized as follows.

1) For the turn-on case:
   a) The diffusion time; *i.e.*, the maximum time required by holes to diffuse across the base region. The maximum diffusion time is calculated from the relation [12]

$$t_D = \frac{2.43}{\omega_{\alpha_0}}. \quad (6)$$

   b) The $r_{bb}C_c$ time constant. (r_{bb} is the base spreading resistance, and C_c is the collector capacitance.)
   c) The response of the emitter junction; *i.e.*, the time it takes for carriers from the emitter to modulate the base so as to increase its conductivity [10].
   d) Premature saturation effects [11]; *i.e.*, forward biasing of the collector base junction at low collector voltages because of the voltage drop developed across the base region by the large base currents.
2) For the turn-off case:
   a) The diffusion time.
   b) The residual storage time [5]; *i.e.*, the time rerequired by the charge stored in the outer recesses of the base to diffuse across the junction.

It should be noted that the concepts of the transistor as a charge-controlled current source have been introduced to discuss transistor transient behavior. Under dc conditions, when we wish to maintain a specific charge distribution in the base, the transistor may be considered to be a current-controlled current source, the control current being that which is required to replace the charge which recombines.

Switching Efficiency

The charge required to turn off a transistor within a specific time has been derived in Appendix II. The result of this analysis is summarized below.

$$(\Delta Q_{\text{off}})_{\min} = Q_s + \bar{C}_c V_c, \quad (7)$$

where Q_s is the value of $Q_b(t)$ in saturation, and $\bar{C}_c$ is the average value of collector capacitance over the voltage swing, as defined in Appendix II.

Since the circuitry is activated and cut off by means of quantities of charge, a convenient descriptive figure of merit of a semiconductor switch is the switch efficiency E_s, which we hereby define by

$$E_s \triangleq \frac{\text{Active region dc control charge}}{\text{Charge required to switch on and switch off}}. \quad (8)$$

The switch-off efficiency is therefore

$$E_s \triangleq \frac{\text{Active region dc control charge}}{\text{Charge required to switch off}}. \quad (9)$$

Let us consider some special cases to determine the limits on E_s. Unlike most commonly defined efficiency quantities, E_s can have values between zero and ∞. Consider for example $(E_s)_{\text{off}}$ for a nonsaturated transistor which we wish to turn off for as short a time as possible (*e.g.*, the diffusion time). Since the transistor is not saturated, the charge stored is $1.22\ I_c/\alpha_0\omega_{\alpha_0}$. Moreover, since the turn-off time is to be very short ($t_{\text{off}}\ll\tau_p$), negligible charge will recombine

$$\int_0^{t_{\text{off}}} \frac{Q_b}{\tau_p}\,dt \approx 0,$$

and the charge which must be inserted is therefore

$$(\Delta Q_{\text{off}}) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \bar{C}_c V_c. \quad (10)$$

Therefore,

$$E_{s_{\text{off}}} = \frac{1}{1 + \dfrac{\omega_{\alpha_0}\bar{C}_c V_c\alpha_0}{1.22I_c}}, \quad (11)$$

but $V_c/I_c = R_L$,

$$E_{s_{\text{off}}} = \frac{1}{1 + \dfrac{\omega_{\alpha_0}\bar{C}_c R_L\alpha_0}{1.22}}. \quad (12)$$

For the conditions

$$\frac{\omega_{\alpha_0}\bar{C}_c R_L\alpha_0}{1.22} \ll 1,$$

$$E_{s_{\text{off}}} \approx 1.$$

The switch-off efficiency of a nonsaturated transistor will thus usually be close to unity.

Now let us calculate the switch-off efficiency of a non-

saturated transistor when $t_{off} \gg \tau_p$; *i.e.*, the turn-off time is long in comparison to the lifetime.

Since all the charge stored in the base will recombine,

$$\Delta Q_{off} = \bar{C}_c V_c,$$

$$E_s = \frac{1.22 I_c}{\bar{C}_c \alpha_0 \omega_{\alpha_0} V_c} = \frac{1.22}{\alpha_0 R_L \bar{C}_c \omega_{\alpha_0}}. \tag{13}$$

For the case

$$\alpha_0 R_L \bar{C}_c \omega_{\alpha_0} \ll 1, \qquad E_{s_{off}} \gg 1.$$

$E_{s_{off}}$ thus reflects the fact that in order to cut a transistor off slowly, it is only necessary to provide the charge required to drive the collector capacity.

Saturated Grounded-Emitter Switches

Four common types of saturating grounded-emitter switches are shown in Fig. 3. The operation and design of these circuits has been extensively discussed in the literature and will only be briefly reviewed here [13], [14].

The Type-1 switch has no reactive components. When the input voltage is at about ground potential, resistors R_b and R_d form a voltage divider with voltage supply E_1 to cut off the base emitter junction of the transistor, Q. With Q at cut-off, the collector load resistor, R_L, can pull down the load to the voltage at which the clamp diode, D_1, will become activated, and the lower voltage level of the circuit will therefore be at $-E_c$. When the input voltage is at $-v$ volts, resistor R_b provides sufficient current to drive resistor R_d and the base of Q until Q saturates and its collector voltage is a few tenths of a volt below ground. The predominant design conditions on R_b and R_d are ac in nature; namely, R_d must provide the amount of overdrive to Q which is required to shut Q off within the specified turn-off time of the switch. Similarly, R_b must provide sufficient overdrive to R_d and the base of Q to saturate Q within the specified turn-on time of the switch. If reasonably fast turn-off and turn-on times of the switch are required, it is usually found that the impedance level of the input circuit and the gain of the switch are quite low.

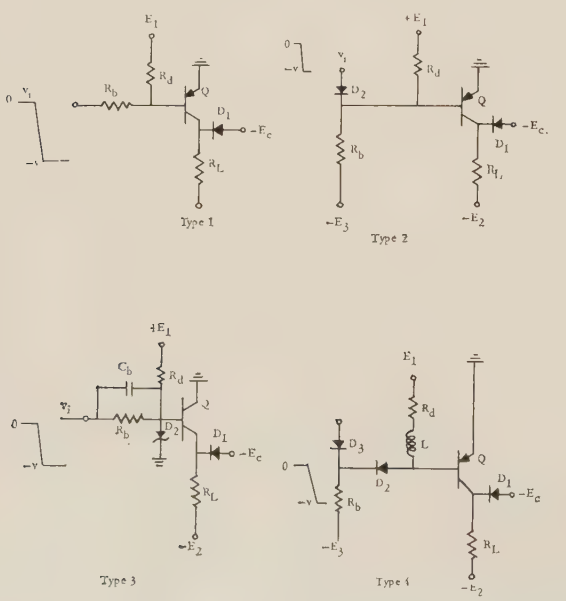


Fig. 3—Four common types of saturated switches.

The Type-2 switch is the current mode analog of the Type-1 switch. When the input is at about ground potential, the current source E_1/R_d is available to drive the base of Q off and maintain Q at cut-off. When the input voltage is $-v$, D_2 becomes reverse biased and the current source $-E_3/R_b$ drives E_1/R_d and the base of Q, thus turning Q on. Both Type-1 and Type-2 switches have approximately the same switching efficiencies.

The Type-3 switch is an RC-coupled inverter. When the input voltage is high, resistors R_d and R_b and supply voltage E_1 form a voltage divider which maintains transistor Q at cutoff. The collector at Q is clamped at $-E_c$. When the input voltage is $-v$, R_b pulls down R_d and supplies the recombination current to the base of Q. C_b couples the charge, $C_b V$, into the base during turn-off and out of the base during turn-on. The steady-state switching efficiency of the Type-3 switch is quite superior to that of the Type-1 and Type-2 switches since the dc base current is maintained at its maximum recombination value rather than being maintained at the value dictated by the switching time requirements. Diode D_2 may be necessary if the $r_{bb}C_b$ time constant becomes comparable to the period of the switch.

The Type-4 switch shown in Fig. 3 is the current mode analog of the Type-3 switch. When the input voltage is 0, current source E_1/R_d provides the I_{co} to maintain Q at cut-off. Current source $-E_3/R_b$ provides the recombination current to maintain Q in the on condition. The inductance allows the full current $-E_3/R_b$ to flow from the base of Q during turn-on and provides the reverse current to turn off Q. Unfortunately, the amount of voltage that can be supported across L, and therefore the initial magnitude of the reverse current, is limited by the base-emitter drop of Q. The reverse current overdrive, and therefore the turn-off time, is a function of the variations in the base-emitter drop of Q. The Type-4 switch may offer some slight advantages over the Type-2 switch.

Nonsaturated Grounded-Emitter Switches

Nonsaturated grounded-emitter switches employ additional circuitry which prevent the base-to-collector junction from becoming forward biased, thereby limiting the charge stored in the base under steady-state conditions to its active region value. However, as will be shown below, the transient charge may frequently exceed the steady-state charge under various conditions of load and drive.

The three most popular techniques which will prevent the base collector diode from becoming forward biased are depicted schematically in Fig. 4. Fig. 4(a) shows the "catcher diode" technique which is one of the best-known

and most commonly used means for preventing saturation. The diode D_1 prevents the collector of the transistor Q_1 from rising appreciably above the upper clamp voltage E_K. When the collector voltage reaches the clamp voltage E_K, the collector load becomes the very low impedance of diode D_1. Since the collector current which will flow is βi_{in}, the power dissipated will be $\beta i_{in} E_K$. The dc power dissipation is thus a sensitive function of β (a parameter which is usually not controlled within narrow limits) and may vary widely. The "catcher diode" technique is highly unreliable, and experience has demonstrated that it results in the frequent destruction of both diodes and transistors. Hence, it will not be considered seriously as an antisaturation technique for a grounded-emitter switch.

The antisaturation circuits schematically depicted in Fig. 4(b) and (c) make use of nonlinear feedback between collector and base to prevent the base-collector diode from becoming forward biased. The base-to-collector voltage in both circuits [Fig. 4(b) and (c)] is determined from the equation

$$V_{eb} + E_B = V_{eb} + V_{bc} + V_{D_1}. \tag{14}$$

Therefore,

$$V_{bc} = E_B - V_{D_1}. \tag{15}$$

Thus the base-collector junction will remain reverse biased providing

$$E_B > V_{D_1}.$$

The collector-to-emitter voltage is therefore

$$V_{ec} = V_{eb} + E_B - V_{D_1}. \tag{16}$$

In practical circuits, the batteries E_B are implemented by voltage dividers, Stabistors, or zener diodes.

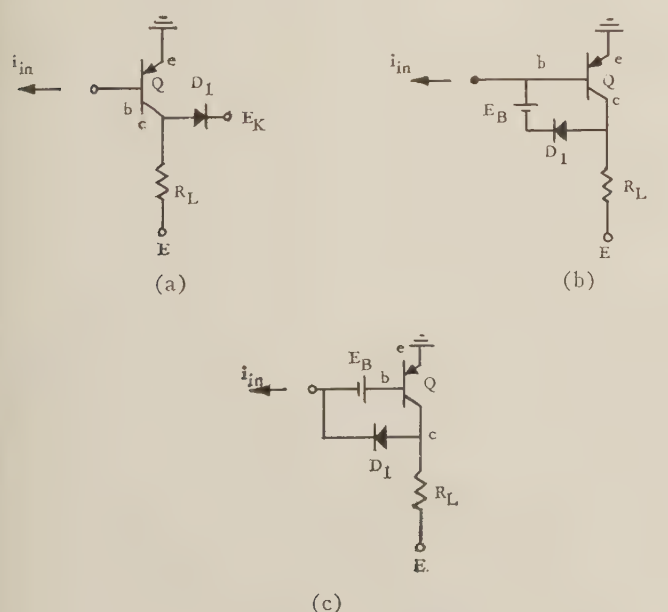


Fig. 4—Antisaturation circuit techniques.

The feedback circuits shown in Fig. 4(b) and (c) avoid the occurrence of a large collector current at a large voltage as in the "catcher diode" technique of Fig. 4(a) and, therefore, are more reliable.

The circuit of Fig. 5 demonstrates the nonsaturation technique of Fig. 4(b). Here, E_B is replaced by the zener diode Z and the current source E_S/R_S.

The nonsaturated analogies of the four switches of Fig. 3 are shown in Fig. 6. It is observed that the theoretical battery E_B shown in Fig. 4(c) has been replaced by a diode D_2. The diode D_2 may, in practice, consist of one or more germanium diodes, silicon diodes, or a zener

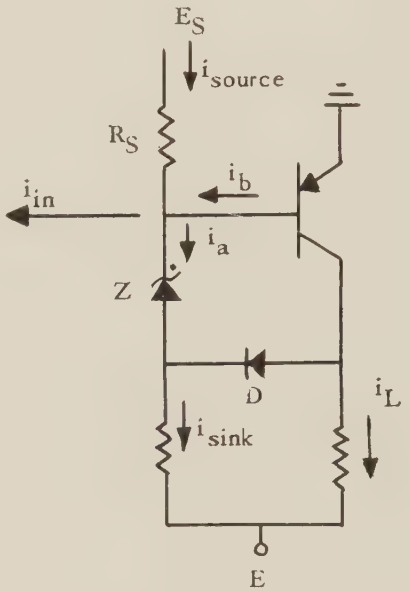


Fig. 5—Antisaturation circuit of Fig. 4(b) mechanized with a zener diode.

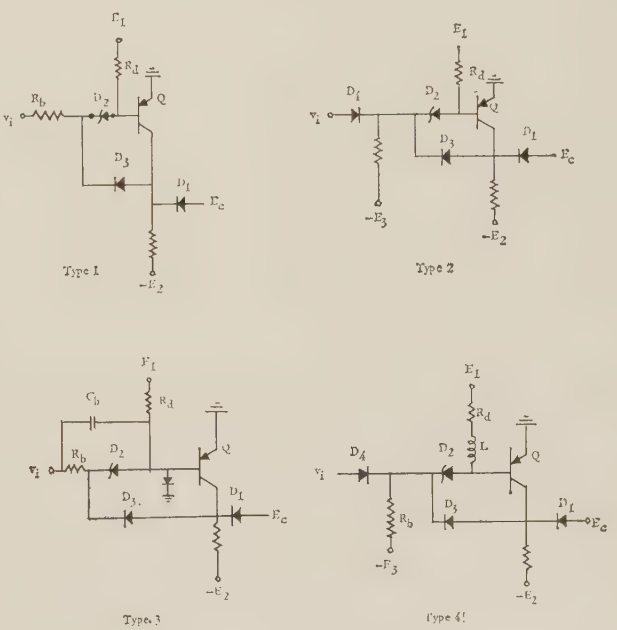


Fig. 6—Nonsaturated switches.

diode, depending on the requirements dictated by the base-to-emitter drop of Q, the environment of the switch, and the tolerance on its grounded-emitter current gain.

Since the voltage E_B is reflected in the collector voltage V_{ec} (16), diode D_2 should exhibit a sharp or high conductance forward characteristic (or reverse characteristic if it is a zener). Care must be taken in the circuit design to insure that D_2 is always operated well beyond the "knee" of the forward characteristic. In the Type-2 and Type-4 nonsaturated switches depicted in Fig. 6, all of the turn-on transient current from R_b must pass through D_2; therefore, D_2 must turn on rapidly or be maintained in the on condition, otherwise, the turn-on characteristics of the switch will be degraded. For example, if E_B is mechanized by a Stabistor (which is notoriously slow in turning on), the degradation in the turn-on response may negate the improvement in the turn-off response. The fast turn-on requirement on D_2 is not a significant problem if D_2 is a germanium diode or a zener. However, it is quite expensive to realize a fast turn-on time if D_2 is a silicon diode.

D_3 should have a good forward regulation characteristic for the same reasons as discussed above. In addition, D_3 should have a low transition region capacity, since this capacity will add directly to the collector capacity of the transistor switch. D_3 should have a very fast reverse recovery time in comparison to the transistor; otherwise, the recovery time of diode D_3 will lengthen the over-all turn-off time of the switch.

Comparison of Saturated and Nonsaturated Circuits

Saturated and nonsaturated circuits may be compared on the basis of five circuit criteria:

1) switchoff efficiency,[2]
2) transient waveforms,
3) stability of voltage levels,
4) power dissipation,
5) noise rejection and suppression ability,
6) circuit complexity and cost.

The following discussion will be an evaluation of the relative merits of the two techniques based on the above criteria; over-all properties of the techniques will be emphasized rather than the properties of specific circuits.

A. Switch-Off Efficiency

All the nonsaturated circuits shown in Fig. 6 are based on the principle that the charge stored in the base region for a given dc collector current can be kept at its active region value merely by preventing the base collector diode from being forward biased; however, the prevention of saturation by this technique is valid only under dc conditions. The limitations of nonsaturated circuitry are particularly evident when:

1) the load in the collector is reactive;
2) the switch is driven from a low impedance source such as a coupling capacitor.

In both of the above situations, the charge stored in the base at the conclusion of the switch-on interval can, or will, exceed the charge required to support the dc current. Consequently, the transistor will be "saturated" despite the presence of the antisaturation circuitry. The charge which will be stored in the base at the conclusion of the turn-on interval under condition 1) will be given by

$$Q_b(T_{on}) = \Delta Q_{on} - Q(\text{recomb}) - Q(C_c), \qquad (17)$$

where

ΔQ_{on} is the charge inserted,
Q (recomb) is the charge which recombines during the turn-on interval,
$Q(C_c)$ is the charge which drives the collector capacity.

The maximum charge stored at T_{on} is, of course,

$$Q_b(T_{on})\Big|_{max} = \Delta Q_{on}. \qquad (18)$$

The charge which will be stored at the conclusion of the turn-on interval for the case of an RC load may be computed with the aid of Fig. 7, assuming that the upper voltage level of the swing is about 0v, where $V_{ce} \cong 0$ and $\omega_\alpha = \omega_{\alpha_0}$. It is observed that there are two components of the charge in the base: the charge Q_1 required to sustain the steady-state dc collector current I_c, and the charge Q_x required to sustain the current which was charging the load capacity just before the transistor bottomed,

$$\left(C_L \frac{dv_c}{dt}\Big|_{0v}\right).$$

Since

$$Q_1 = \frac{1.22}{\alpha_0 \omega_{\alpha_0}} I_c, \qquad (19)$$

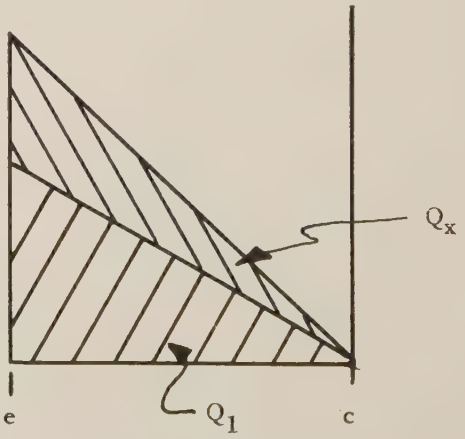


Fig. 7—Base charge distribution at end of rise time with capacitive load.

[2] Saturated and nonsaturated switches exhibit the same switch-on efficiencies.

$$Q_x = \frac{1.22}{\alpha_0\omega_{\alpha_0}} C_L \frac{dv_c}{dt}\bigg|_{0v}. \tag{20}$$

Therefore,

$$Q_b(T_{on}) = Q_1 + Q_x = \frac{1.22}{\alpha_0\omega_{\alpha_0}}\left\{I_c + C_L \frac{dv_c}{dt}\bigg|_{0v}\right\}. \tag{21}$$

The excess charge

$$\frac{1.22}{\alpha_0\omega_{\alpha_0}} C_L \frac{dv_c}{dt}\bigg|_{0v}$$

will decay with a time constant K_s [as described in (51) of Appendix I]. Only after this transient excess charge has decayed will the nonsaturated circuitry become effective. Clearly, the effectiveness of the nonsaturated circuitry is related to the frequency at which the switch is operated. The limitations of the nonsaturated switches under the above conditions may be demonstrated by comparing the switch-off efficiency of the nonsaturated switch with its saturated counterpart.

It has been shown that the charge which will be stored in the base at the conclusion of the turn-on interval for an RC load [case 1)] is

$$Q_b(T_{on}) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \frac{1.22C_L}{\alpha_0\omega_{\alpha_0}}\left\{\frac{dv_c}{dt}\right\}_{0-}$$

and that the maximum charge stored in the base if the transistor is driven by a low impedance charge source (or) [case 2)] is

$$Q_b(T_{on}) = \Delta Q_{on}. \tag{18}$$

For case 2):

$$\Delta Q_{off}(T_{on} + \Delta t) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\Delta Q_{on} - \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}}\right\} e^{-\Delta t/K_s} + K_s\left[I_b - \frac{I_c}{\beta}\right][1 - e^{-\Delta t/K_s}] + \bar{C}_c\Delta v_c. \tag{24}$$

If the switches are not saturated, $I_b = \beta I_c$, and we obtain for case 1):

$$\Delta Q_{off}(T_{on} + \Delta t) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\frac{1.22C_L}{\alpha_0\omega_{\alpha_0}} \frac{dv_c}{dt}\bigg|_{0v}\right\} e^{-\Delta t/K_s} + \bar{C}_c\Delta v_c. \tag{25}$$

For case 2):

$$\Delta Q_{off}(T_{on} + \Delta t) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\Delta Q_{on} - \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}}\right\} e^{-\Delta t/K_s} + \bar{C}_c\Delta v_c. \tag{26}$$

The ratio of the switch-off efficiencies of the saturated and nonsaturated switches is

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}} = \frac{\dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}}}{\dfrac{(\Delta Q_{off})\ \text{nonsat}}{1.22I_c}} = \frac{(\Delta Q_{off})\ \text{sat}}{(\Delta Q_{off})\ \text{nonsat}}. \tag{27}$$

For case 1):

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}} = \frac{\dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\dfrac{1.22C_L}{\alpha_0\omega_{\alpha_0}} \dfrac{dv_c}{dt}\bigg|_{0v}\right\} e^{-\Delta t/K_s} + K_s\left[I_b - \dfrac{I_c}{\beta}\right][1 - e^{-\Delta t/K_s}] + \bar{C}_c\Delta v_c}{\dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\dfrac{1.22C_L}{\alpha_0\omega_{\alpha_0}} \dfrac{dv_c}{dt}\bigg|_{0v}\right\} e^{-\Delta t/K_s} + \bar{C}_c v_c}. \tag{28}$$

For case 2):

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}} = \frac{\dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\Delta Q_{on} - \dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}}\right\} e^{-\Delta t/K_s} + K_s\left[I_b - \dfrac{I_c}{\beta}\right][1 - e^{-\Delta t/K_s}] + \bar{C}_c\Delta v_c}{\dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left[\Delta Q_{on} - \dfrac{1.22I_c}{\alpha_0\omega_{\alpha_0}}\right] e^{-\Delta t/K_s} + \bar{C}_c\Delta v_c}. \tag{29}$$

The charge required to turn off the switch at time $T_{on}+\Delta t$ for the two above conditions may be calculated from (21), (18), and (51).

For case 1):

$$\Delta Q_{off}(T_{on} + \Delta t) = \frac{1.22I_c}{\alpha_0\omega_{\alpha_0}} + \left\{\frac{1.22C_L}{\alpha_0\omega_{\alpha_0}} \frac{dV_c}{dt}\bigg|_{0v}\right\} e^{-\Delta t/K_s} + K_s\left[I_b - \frac{I_c}{\beta}\right][1 - e^{-\Delta t/K_s}] + \bar{C}_c\Delta v_c. \tag{23}$$

Eqs. (28) and (29) clearly indicate that the relative improvement in switch-off efficiency gained by the use of nonsaturated circuitry is a function of the frequency with which the switch is operated, *i.e.*, the time interval Δt which elapses between turn-on and turn-off.

As $\Delta t/K_s \to 0$,

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}} \to 1 \quad \text{for both the above cases.}$$

As $\Delta t/K_s \to \infty$,

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}} = \frac{\dfrac{1.22I_c}{\omega_{\alpha_0}\alpha_0} + K_s\left[I_b - \dfrac{I_c}{\beta}\right] + \bar{C}_c v_c}{\dfrac{1.22I_c}{\omega_{\alpha_0}\alpha_0} + \bar{C}_c v_c}. \quad (30)$$

Hence, it is quite evident that for frequencies such that $\Delta t/K_s \lesssim 1$, very little improvement in switching efficiency results from the application of anitisaturation circuitry. The equations derived above may be used to determine quantitatively the relative improvement in switch-off efficiency which occurs with the application of nonsaturated circuitry. Fig. 8 shows a plot of

$$\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}}$$

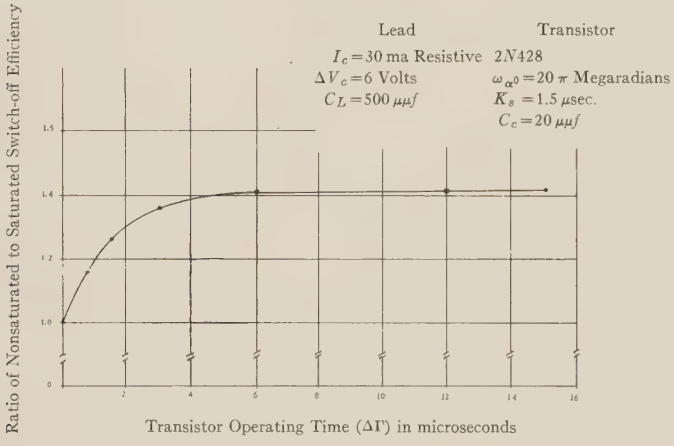


Fig. 8—Graph of ratio of nonsaturated to saturated switchoff efficiency vs transistor operating time.

for a 2N428 driver as a function of Δt. The curve was computed for the following transistor specifications:

$$\beta_{min} = 100,$$

$$\beta_{max} = 300,$$

$$\omega_{\alpha_0} = 20\pi \text{ megaradians},$$

$$K_s = 1.50\ \mu\text{sec}.$$

Under the steady-state conditions,

$$\left.\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}}\right|_{max} = 1 + \frac{K_{s_{max}}\left[\dfrac{I_b}{I_{c_{max}}} - \dfrac{1}{\beta_{max}}\right]}{\dfrac{1.22}{\alpha_0\omega_{\alpha_{0_{max}}}} + \bar{C}_{c_{min}}\left(\dfrac{V_c}{I_c}\right)}. \quad (30a)$$

However, the dc circuitry must be designed such that

$$I_b > \frac{I_{c_{max}}}{\beta_{min}}.$$

Therefore,

$$\frac{I_b}{I_{c_{max}}} \geq \frac{1}{\beta_{min}}$$

and

$$\frac{V_c}{I_c} = R_L,$$

resulting in

$$\left.\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}}\right|_{max} = 1 + \frac{(K_s\omega_{\alpha_0})_{max}\left[1 - \dfrac{\beta_{min}}{\beta_{max}}\right]}{\left\{\dfrac{1.22}{\alpha_0} + R_L\bar{C}_{c_{min}}\omega_{\alpha_{0_{max}}}\right\}\beta_{min}}.$$

Thus, the maximum possible improvement in steady-state switch-off efficiency is a function of ω_{α_0}, K_s, β_{min}, and the β spread of the transistor. Adequate specifications on these device parameters may result in only a slight difference in switch-off efficiency between saturated and nonsaturated operations under steady-state conditions.

For a 2N428 with a specified maximum 3:1 β spread,

$$\left.\frac{(E_{s_{off}})\ \text{nonsat}}{(E_{s_{off}})\ \text{sat}}\right|_{max} = 1.4.$$

The previous discussion has shown that nonsaturated circuits are only completely effective when $(\Delta t/K_s) \gg 1$, where Δt is the time interval between turn-on and turn-off of the switch. Moreover, it has been shown that through judicious establishment of transistor specifications, the steady-state saturated switch efficiency may be made to approach the nonsaturated switch efficiency. The techniques established above may be extended to compare the relative advantages of saturated and nonsaturated techniques for various types of loads and drives.

B. Waveforms

One of the most serious limitations on nonsaturated circuits is the severe distortion that they introduce into the turn-on waveform for certain conditions of drive and load. The distortion can be attributed to the lag and resulting over-compensation in the feedback loop. This lag is due to the fact that the nonsaturated circuitry can only prevent the base-collector junction from becoming forward biased; it cannot however prevent the charge stored in the base from exceeding its dc level when the load is capacitive or when the transistor is driven from a low impedance source. Fig. 9 shows a nonsaturated switch (Fig. 6, Type 1) driving a predominately capacitive load. Fig. 10 shows the typical voltage response of the switch for the input voltage step shown.

Upon the application of an input voltage step of $-V$, a base current step of

$$\frac{V - V_{eb} - V_{D_2}}{R_b}$$

is steered into the base of Q. Diode D_3 is reverse biased during this interval. When the collector voltage reaches the value $-V_{eb} - V_{D_2}$, diode D_3 becomes forward biased.

owever, the charge stored in the base region is con-lerably in excess of that which is required to sustain e dc current E_2/R_L, and the collector voltage con-nues to increase. The forward biasing of diode D_3 al-ws a large current to pass through D_3, and the input rrent, rather than being routed into the base of Q, is ncelled by the current through D_3. The reverse cur-nt source E_1/R_d is steered into the base of Q, and ode D_2 is effectively cut off. During this interval, the ansistor is saturated and its collector-to-emitter volt-e remains at a low value. The reverse current E_1/R_d, d recombination, eventually cause the charge in the se to decay to the value required to sustain the dc rrent E_2/R_L. However, since the base current has re-rsed, there is no current to replenish the charge lost in combination, and the charge falls below the value

$$\frac{1.22}{\alpha_0\omega_{\alpha_0}} \frac{E_2}{R_L}.$$

s a result, the collector voltage begins to fall. The ecay of the collector voltage will result in D_3 being t off, and the input current will again drive E_1/R_d and e steered back into the base of Q. The charge will then uild up to support the dc collector current and the oltage will eventually stabilize at its nonsaturated alue. The magnitude of the dip will depend upon C_L, $_L$, and E_1/R_d, and the properties of the transistor; it ay be computed using circuit analysis techniques in onjunction with the charge equilibrium equations de-ved in Appendix II. The phenomena which occur at various times have been noted on the waveform of Fig. 10.

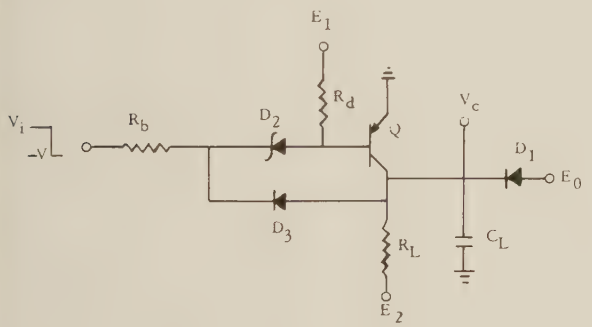


Fig. 9—Type-1 nonsaturated switch driving a capacitive load.

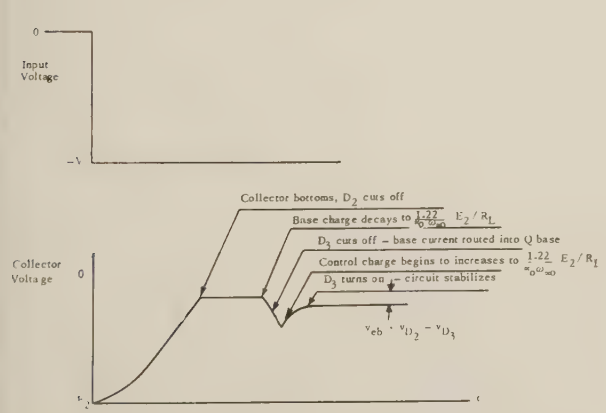


Fig. 10—Voltage response of the transistor switch of Fig. 9.

The same type of waveform occurs when a Type-3 switch (Fig. 6) is used, even though the load is purely resistive. In the case of a Type-3 switch, the charge deposited into the base is approximately $C_b\Delta V$ (where ΔV is the voltage step at the input). The transistor will remain saturated until the base charge decays to that required to sustain the dc current. The excess charge will decay with a time constant K_s, as has been noted above. The occurrence of the large noise pulses caused by the nonsaturated switch are unacceptable in certain types of computer logic circuits. It may therefore be necessary to introduce additional circuitry to limit the magnitude of the noise pulses or to make succeeding circuitry insensitive to them.

It should be mentioned that under the conditions of high impedance drives (Fig. 6, switch Types 1, 2, and 4) and with resistive loads, the voltage waveforms are quite distortionless and resemble those which one observes with saturated circuitry.

C. Stability of Voltage Levels

A most significant problem in the application of a nonsaturated switch is presented by the rather poor stability of the voltage level when the transistor is on. The collector-to-emitter voltage of the circuit of Fig. 4(c) is strongly dependent on input current, load current, transistor β, and ambient temperature. Moreover, it can be observed that the output voltage will reflect the tolerances on the transistor base input characteristic and D_1, D_2 diode characteristics. Although the temperature variations of the base-emitter characteristic and diode D_3 will tend to cancel (if these diodes are of the same material), the collector voltage will reflect the thermal variation of the diode D_2 and the temperature dependence of β.

It will be valuable for the purposes of this paper to compare the experimental results of the dc performance of the circuit of Fig. 9 with that of a saturated transistor operating under the same conditions. Figs. 11 and 12 show a plot of V_{ec} vs I_c as a function of temperature for the Type-1 circuits of Figs. 3 and 6. Q was a 2N428 transistor having a minimum β specification of 60 at 60 ma, D_2 was a SG22 Stabistor, and D_3 a 1N283 diode. I_{in} was set at 1 ma and 3 ma; no reverse current was supplied. The maximum voltage variation for the nonsaturated transistor over the complete range of operation conditions ($5\text{ ma} \leq I_c \leq 50\text{ ma}$; $1\text{ ma} \leq I_{in} \leq 3\text{ ma}$; $-15°\text{C} \leq T \leq 60°\text{C}$) was 0.56 volt, while that for the saturated switch was 0.13 volt. The nonsaturated transistor exhibited four times the voltage variation of the saturated transistor. Figs. 11 and 12 dramatically show the difference in the voltage stability of the saturated and nonsaturated switches.

In the practical case, the nonsaturated switch will exhibit even larger variations compared to the saturated switch because of the tolerances on the forward charac-

teristic of diodes D_2 and D_3 and the variation of β. The larger collector voltage variations which result with nonsaturated circuitry dictate larger logical swings and result in greater dc power dissipation in computer circuitry.

D. Power Dissipation

It is well recognized that nonsaturated transistors will exhibit greater power dissipation than their saturated counterparts because of the higher collector voltage. However, the strong dependence of the power dissipated on such factors as switch-on input current, load current, transistor β spread and ambient temperature, is often overlooked.

The higher power dissipation of the nonsaturated transistor and its greater sensitivity to operating conditions and ambient temperature are demonstrated by power dissipation curves of Figs. 13 and 14. The power dissipated by the nonsaturated transistor at 60°C and 50 ma is five times that dissipated by the saturated transistor. Furthermore, its rate of increase of power dissipation with collector current is about 4.5 times that for the saturated transistor.

The above measurements and computations have

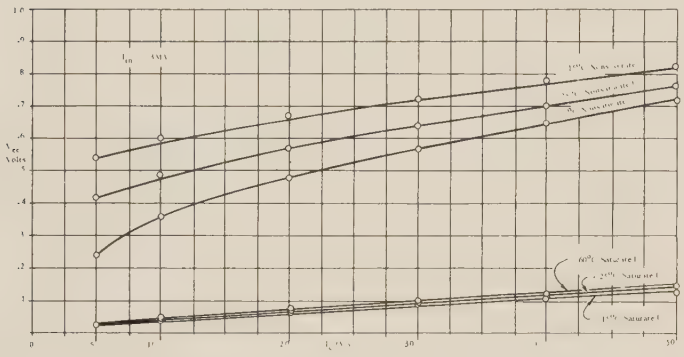


Fig. 11—V_{ec} vs I_c as a function of temperature for a high β 2N428.

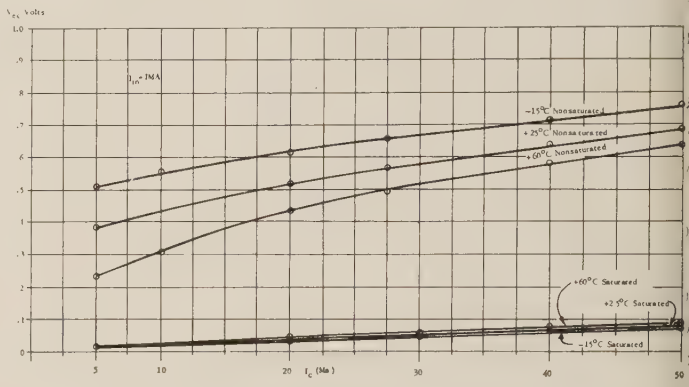


Fig. 12—V_{ec} vs I_c as a function of temperature for a high β 2N428.

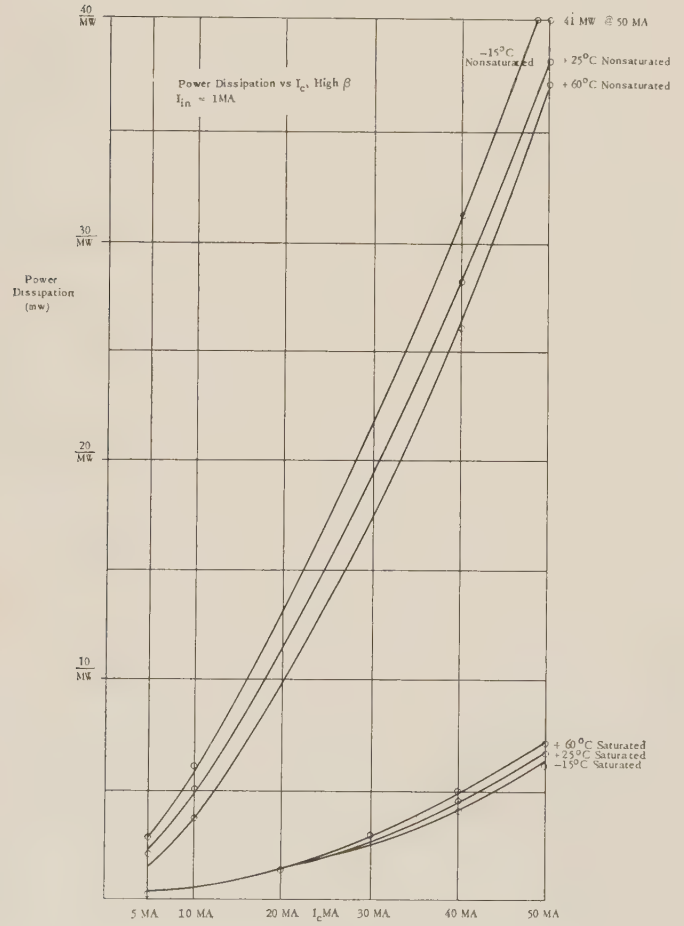


Fig. 13—Power dissipation curves.

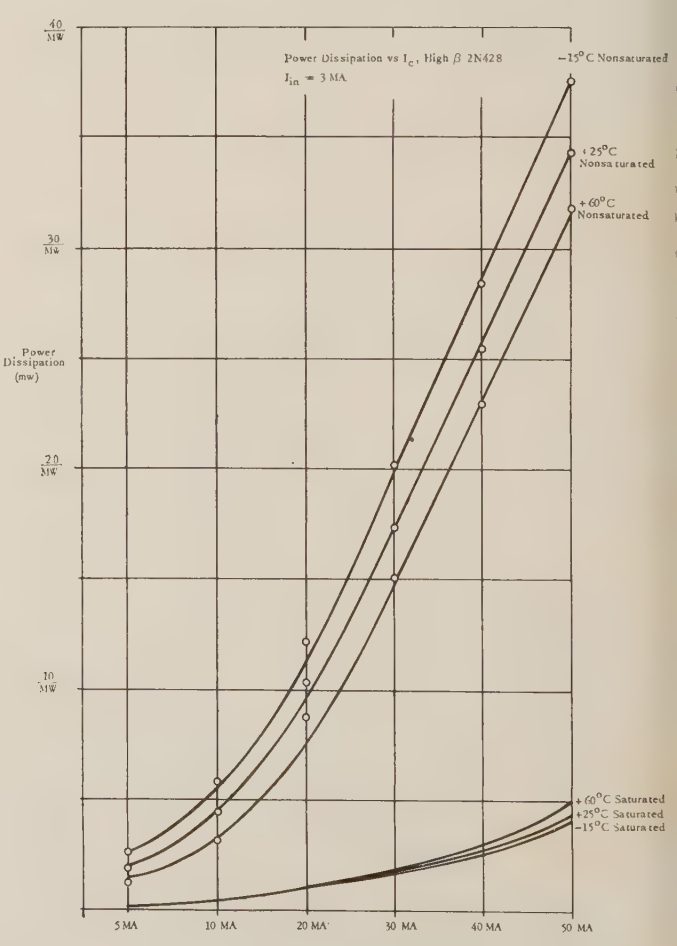


Fig. 14—Power dissipation curves.

been made to provide some concept of the typical relative power dissipations which will be experienced with saturated and nonsaturated switches for a specific transistor. It should also be noted that the power dissipation curve for the nonsaturated transistor will reflect the variation in β and the tolerances on diodes D_3 and D_2.

E. Noise Rejection and Suppression Ability

Always of importance in the selection of a switch for a computing system is its ability to reject and suppress unwanted transient voltages.

The noise rejection ability of the several types of switches differs. The voltage mode switches are less sensitive to noise than the current mode switches; the dc switches are less sensitive than the switches containing ac elements. The switches may be rated in order of decreasing ability to reject noise as follows: Type 1, Type 2, Type 3, Type 4.

Although both saturated and nonsaturated switches are equally sensitive to noise voltages which will tend to turn them on (negative pulses), the saturated switches are less susceptible to turn-off noise. This rejection ability results from the fact that the saturated transistor stores more control change than is actually required to sustain a given collector current. Hence, it can withstand a certain perturbation of this control charge without affecting its collector current.

The ability of a switch to suppress noise is related to the impedance it presents to the noise in the "on" condition.

Fig. 15 shows a typical system of switching circuits. Transistor T_{R_2} is driven by T_{R_1}. T_{R_1} is normally on and T_{R_2} normally cut off. e_n is a noise generator which couples negative voltage spikes to line l, tending to turn T_{R_2} on. The noise developed at x will depend on the ratio of the impedance of the line and the impedance offered by T_{R_1}. Therefore,

$$V_x = \frac{e_n}{1 + [Z(l)/Z(T_{R_1})]}.$$

Since a saturated transistor will present a lower impedance to ground than a nonsaturated transistor, it will insure a smaller voltage excursion at point x. Consequently, a saturated switch is a better noise suppressor than a nonsaturated switch because of its lower resistance in the "on" condition.

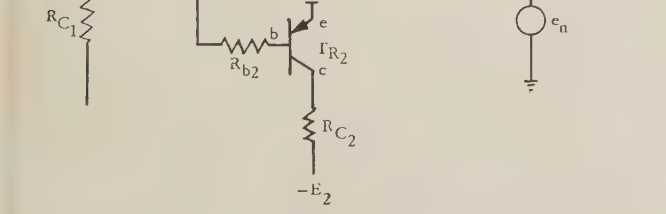


Fig. 15—Typical system of switching circuits.

F. Circuit Complexity and Cost

Examination of the several types of saturated and nonsaturated circuits of Figs. 3 and 6 indicates that (excluding the unreliable "catcher diode" techniques) the nonsaturated circuits require at least two components to prevent dc saturation. Although the inclusion of two additional components may not create any significant packaging problems, the cost of the additional circuitry may prove to be a significant portion of the cost of the saturated switch. For example, in the Type-1 nonsaturated switch of Fig. 6, diode D_1 must be a fast recovery diode; therefore, the cost of this diode may be anywhere from \$.50 to \$2.00, depending on whether the diode is made of silicon or germanium.

Diode D_2 is usually a single silicon diode if Q is a germanium transistor, but consists of two (or more) silicon diodes or a zener if Q is a silicon transistor. Since silicon diodes are notoriously slow in turning on, high-speed circuits may demand that D_2 be a very fast silicon diode, or zener. As an alternate choice, D_2 can be bypassed by a capacitor. If this is not done, the improvement in switch-off efficiency may be offset by the degradation in turn-on performance. The former alternatives are expensive, while the latter alternative results in a lower transient switching efficiency.

The significant point is that the additional cost of the nonsaturated circuitry may not justify the increase in switch-off efficiency, even if the switching frequency is such that the nonsaturated circuitry is fully effective. The designer must carefully weigh the cost of the nonsaturated circuitry with its increased switching efficiency against the cost of a more expensive transistor (having a lower K_s) or additional switches.

Conclusions

The preceding discussion has attempted to provide a basis for comparison of saturated and nonsaturated switches. It has been shown that the only significant circuit advantage enjoyed by nonsaturated transistor switches over their saturated counterpart is an increased switchoff efficiency under dc conditions. The value of this increased switch-off efficiency is a function of the frequency of the switch and the K_s of the transistor. The nonsaturated switches have been shown to be inferior to saturated switches in their transient waveforms, voltage stability, power dissipation, noise suppression and rejection ability, and cost. It is impossible to make any sweeping statements as to whether the nonsaturated technique is better or worse than the saturated technique, since the quantitative advantages that one method enjoys over the other are functions of the application at hand. However, it is hoped that this paper has given the circuit designer some insight with which he may intelligently choose between the two techniques.

Appendix I

Calculation of Stored Charge Q_b

Steady-State Case

Fig. 16 shows schematically the minority carrier density distribution in the base of a saturated *p-n-p* transistor where

$$Q_b = Q_1 + Q_{bx}. \tag{31}$$

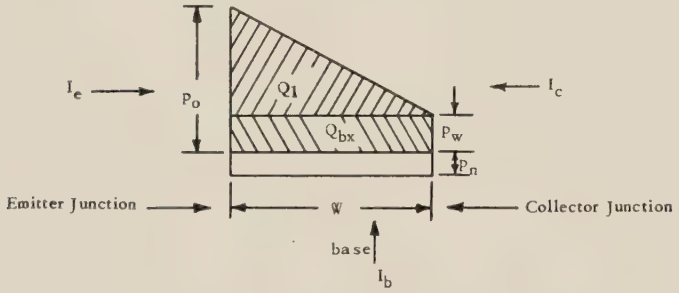


Fig. 16—Minority carrier distribution in the base of a saturated *p-n-p* transistor.

1) *Calculation of* Q_1:

$$Q_1 = \frac{q}{2}(p_0 - p_w)W, \tag{32}$$

but if $I_c \gg I_b$ in the active region,

$$I_e = \frac{qD_p(p_0 - p_w)}{W}. \tag{33}$$

Therefore,

$$p_0 - p_w = \frac{WI_e}{qD_p} \tag{34}$$

and

$$Q_1 = \frac{qW^2I_e}{2 \cdot q \cdot D_p} = \frac{(1.22)I_e}{\omega_\alpha} = \frac{1.22I_c}{\omega_\alpha \alpha_0}. \tag{35}$$

2) *Calculation of* Q_2:

$$Q_{bx} = qp_w \cdot W, \tag{36}$$

but

$$p_w = p_n e^{qV_{cb}/kT}, \tag{37}$$

where V_{cb} is the forward collector-to-base drop. Ebers and Moll [7] have shown that, in saturation,

$$e^{qV_{cb}/kT} = -\left(\frac{I_c + \alpha_0 I_e}{I_{co}}\right) \tag{38}$$

where α_0 is the emitter-to-collector dc current gain measured at the edge of saturation, and I_{co} is the base-to-collector saturation current. Since

$$I_e = -I_b - I_c, \tag{39}$$

$$e^{qV_{cb}/kT} = \frac{-(1-\alpha_0)}{I_{co}}\left(I_c - \frac{\alpha_0}{1-\alpha_0} \cdot I_b\right). \tag{40}$$

Since $\beta_0 = \alpha_0/(1-\alpha_0)$,

$$e^{qV_{cb}/kT} = \frac{1}{I_{co}(1+\beta_0)}(\beta_0 I_b - I_c). \tag{41}$$

Therefore,

$$Q_{bx} = \frac{qp_nW}{I_{co}(1+\beta_0)}(\beta_0 I_b - I_c) \tag{42}$$

and

$$Q_b = Q_1 + Q_{bx} = \frac{1.22I_c}{\alpha_0\omega_\alpha} + \frac{qp_nW\beta_0}{I_{co}(1+\beta_0)}\left(I_b - \frac{I_c}{\beta_0}\right). \tag{43}$$

Defining K_s, the saturation time constant, as

$$K_s = \frac{qp_nW\alpha_0}{I_{co}},$$

we obtain

$$Q_b = \frac{1.22I_c}{\alpha_0\omega_\alpha} + K_s\left(I_b - \frac{I_c}{\beta_0}\right). \tag{44}$$

Transient Case

Consider the circuit of Fig. 17. TR_1 is driven into saturation by a charge impulse $C_b\Delta V$ at $t=t_0$; it is then maintained in saturation by the current $I_b = \Delta v/R_b$. Since the transistor TR_1 is saturated for times $t > t_0$, its collector-to-emitter voltage will remain relatively constant at approximately zero volts; hence, the collector current of TR_1 also will remain constant at about E_s/R_c. It should be noted that, although the collector voltage and collector current of TR_1 are relatively constant for $t > t_0$, the charge stored in the base of TR_1 most certainly is not.

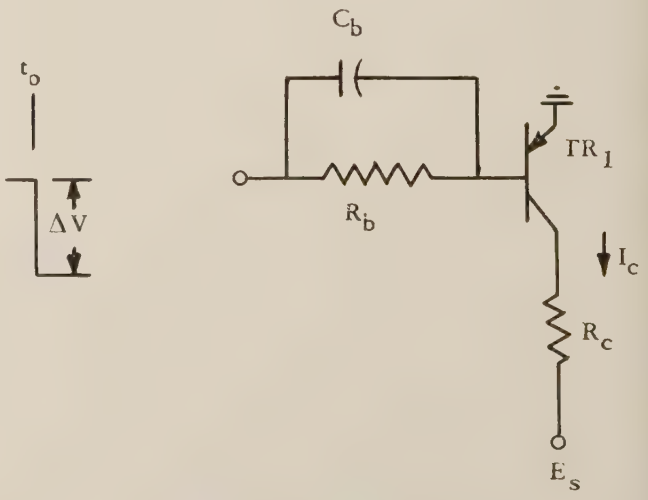


Fig. 17—Saturated driver.

The charge stored in the base region for times $t>t_0$ is mputed below. Let

Q_b be the total charge stored in the base region at any time t,

$Q_b = Q_b(t)$,

Q_1 be the charge required to sustain the collector current I_c, and

Q_{bx} be the excess charge stored in the base region at any time t.

In Fig. 18

r_{bb} is the base spreading resistance,

C_e is the emitter junction transition capacity,

τ_p is the active region base time constant and is given approximately by $\tau_p = 1.22\beta/\omega\alpha$,

K_s is the saturation time constant,

S_b is an infinite capacitance which stores the total base control charge Q_b,

I_{co} is the leakage current of the collector-base diode,

C_c is the collector junction transition capacity, and

τ_c is the active region collector time constant and is given approximately by $\tau = 1.22/\omega\alpha$.

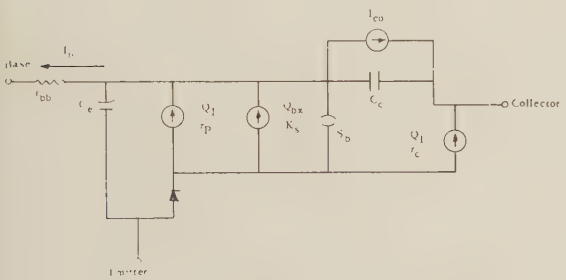


Fig. 18—Saturation region equivalent circuit.

or times greater than t_0, we will assume that

$$C_e \frac{dV_e}{dt} = 0, \tag{45}$$

$$C_c \frac{dV_c}{dt} = 0, \tag{46}$$

nd I_{co} is negligible.

Since I_c is constant, Q_1 is constant and $dQ_1/dt = 0$,

$$I_b - \frac{Q_1}{\tau_p} = \frac{Q_{bx}}{K_s} + \frac{d}{dt} Q_{bx} \tag{47}$$

ut

$$\frac{Q_1}{\tau_p} = \frac{1.22 I_c}{\alpha_0 \omega_\alpha} \cdot \frac{\omega_\alpha}{1.22\beta} \cong \frac{I_c}{\beta}.$$

q. (47) reduces to

$$I_b - \frac{I_c}{\beta} = \frac{Q_{bx}}{K_s} + \frac{d}{dt} Q_{bx}, \tag{48}$$

the solution of which is

$$Q_{bx}(t) = A e^{-t/K_s} + K_s \left[I_b - \frac{I_c}{\beta} \right]. \tag{49}$$

The initial condition on Q_{bx} is

$$Q_{bx}(0) = Q_0 - \frac{1.22 I_c}{\alpha_0 \omega_\alpha}.$$

Upon evaluating A and substituting back into (49), we obtain

$$Q_{bx}(t) = \left[Q_0 - \frac{1.22 I_c}{\alpha_0 \omega_\alpha} \right] e^{-t/K_s} + K_s \left[I_b - \frac{I_c}{\beta} \right] [1 - e^{-t/K_s}], \tag{50}$$

and the total charge is therefore

$$Q_b(t) = \frac{1.22 I_c}{\alpha_0 \omega_\alpha} + \left[Q_0 - \frac{1.22 I_c}{\alpha_0 \omega_\alpha} \right] e^{-t/K_s} + K_s \left[I_b - \frac{I_c}{\beta} \right] [1 - e^{-t/K_s}]. \tag{51}$$

For $t = \infty$,

$$Q_b(\infty) = \frac{1.22 I_c}{\alpha_0 \omega_\alpha} + K_s \left[I_b - \frac{I_c}{\beta} \right],$$

which agrees with the steady-state value of storage charge calculated above.

Appendix II

Calculation of Turn-Off Charge

Introduction

A model of a saturated p-n-p transistor is shown in Fig. 19. The differential equations which account for charge transfers during the transient turn-off interval consist of

$$I_e = I_{e\,\mathrm{diff}} + \frac{Q_b}{\tau_p} + \frac{dQ_b}{dt} + C_e \frac{dv_e}{dt}, \tag{52}$$

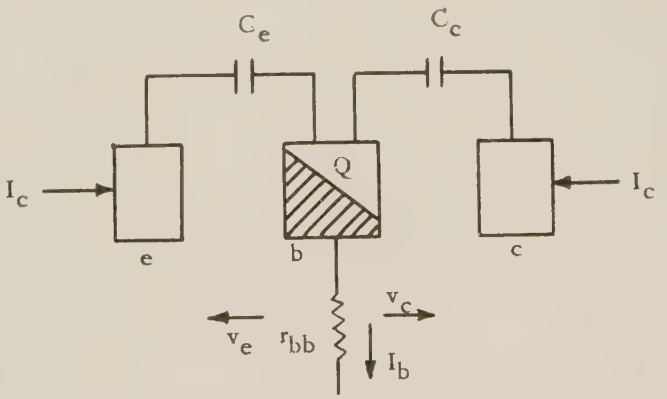


Fig. 19—Model of saturated p-n-p transistor.

$$I_c = -I_{c\,\mathrm{diff}} + C_c \frac{dv_c}{dt}, \tag{53}$$

$$I_b = I_e + I_c = \frac{Q_b}{\tau_p} + C_c \frac{dv_c}{dt} + C_e \frac{dv_e}{dt} + \frac{dQ_b}{d}, \tag{54}$$

where

Q_b/τ_p is the recombination current,
$C_c(dv_c/dt)$ is the current which discharges the collector junction capacity,
$C_e(dv_e/dt)$ is the current which discharges the emitter junction capacity,
dQ_b/dt is the current which discharges the base region.

To switch a transistor from an operating condition in the active or saturation region (where the charge stored is Q_s), to cut-off, we must change the base charge from Q_s to zero, the collector capacity charge by $C_c dv_c$, and the emitter capacity charge by $C_e dv_e$. Eq. (54) may be integrated to determine the charge necessary to cut the transistor off.

$$(\Delta Q_{\mathrm{off}}) + \int_0^t \frac{Q_b}{\tau_p}\,dt = \int_0^{-v_c} C_c dv_c + \int_{-v_e}^{0} C_e dv_e + Q_s \tag{55}$$

where Q_s is the charge which was stored in the base of the transistor at $t=0$ and t is the time required to remove the charge ΔQ_{off}.

$$\Delta Q_{\mathrm{off}} = \int_0^{-v_c} C_c dv_c + \int_{-v_e}^{0} C_e dv_e + Q_s - \int_0^t \frac{Q_b}{\tau_p}\,dt. \tag{56}$$

C_e is usually of the order of 20 $\mu\mu f$ and v_e is approximately 0.3 volt (for germanium transistors). The value of the integral

$$\int_{-v_e}^{0} C_e dv_e$$

is therefore of the order of 6 $\mu\mu$coulombs which is quite small compared to the other terms; hence we will assume that

$$\int_{-v_e}^{0} C_e dv_e$$

may be dropped.

τ_p is the lifetime of the holes in the n region and is consequently a very sensitive function of carrier density, charge distribution, and the conditions at the junction; moreover, the charge which recombines during the turn-off interval

$$\left[\int_0^t \frac{Q_b}{\tau_p}\,dt\right]$$

cannot be determined exactly unless $Q_b(t)$ is accurately calculated, which is an alternative plagued with numerous difficulties. If the turn-off time is short,

$$\int_0^t \frac{Q_b}{\tau_p}\,dt \ll Q_s.$$

The maximum turn-off charge which will be necessary may be computed by neglecting the charge which recombines

$$(\Delta Q_{\mathrm{off}})_{\max} = Q_s + \int_0^{-v_c} C_c dv_c, \tag{57}$$

$$(\Delta Q_{\mathrm{off}})_{\max} = \int_0^t I_b dt = Q_s + \int_0^{-v_c} C_c dv_c. \tag{58}$$

Q_b is the charge which was stored in the base region prior to the application of the reverse base drive;

$$\int_0^{-v_c} C_c dv_c$$

is the charge required to drive the collector capacity through the voltage v_c.

Since $C_c = f(v_c)$, then

$$\int_0^{-v_c} C_c dv_c = \Delta v_c \left[\frac{1}{v_c}\int_0^{-v_c} C_c dv_c\right].$$

But

$$\frac{1}{v_c}\int_0^{-v_c} C_c dv_c = \overline{C}_c,$$

where $\overline{C}_c$ is the average value of capacitance over the voltage swing. This analysis is investigated in more detail elsewhere [4]. The result is

$$\int_0^{-v_c} C_c dv_c = \overline{C}_c \Delta v_c.$$

By time t, sufficient charge has been inserted into the transistor to completely discharge the base and thus cause the transistor to cut off. It is quite possible that by, or before, time t, both the emitter and collector junction will have recovered and no further injection of holes into the base will be possible; however, collector current will continue to flow for an additional time t_D until those holes which remain in the base have diffused out of the transistor through either emitter or collector junctions.

The average time required for holes to diffuse across the base region to either junction is referred to as the diffusion time. The maximum time required to turn off the transistor current source (t_{off}) is therefore

$$t_{\mathrm{off}} = t_D + t, \tag{59}$$

where t is calculated from the relation

$$\int_0^t I_b dt = Q_s + \overline{C}_c \Delta v_c. \tag{60}$$

should be noted that the charge required to turn a nsistor off is independent of the terminal conditions the transistor; the turn-off time is dependent only on the speed with which the charge can be made ilable at the base and the transistor diffusion time.

Bibliography

R. Beaufoy and J. J. Sparkes, "The junction transistor as a charge controlled device," *ATE J.*, vol. 13, pp. 310–327; October, 1957.
R. D. Lohman, "Charge storage in alloyed junction transistors," RCA Rept., May, 1957.
D. Dewitt and A. L. Rossoff, "Transistor Electronics," McGraw-Hill Book Co., Inc., New York, N. Y.; 1957.
G. H. Goldstick, "The Synthesis of Alloy Junction Transistor Switching Circuits," M.S. thesis, Dept. of Engrg., University of California, Los Angeles; September, 1958.
N. F. Moody, "Controlled saturation in transistors and its application in trigger circuit design," *Electronic Engrg.*, vol. 30, pp. 121–127; March, 1958.
C. D. Simmons, "Hole storage delay time and its prediction," *Semiconductor Products*, vol. 1, pp. 14–18; May–June, 1958.
[7] J. J. Ebers and J. L. Moll, "Large signal behavior of junction transistors," Proc. IRE, vol., 42, pp. 1761–1772; December, 1954.
[8] J. L. Moll, "Large signal transient response of junction transistors," Proc. IRE, vol. 42, pp. 1773–1784; December, 1954.
[9] J. W. Easley, "The effect of collector capacity on the transient response of junction transistors," IRE Trans. on Electron Devices, vol. ED-4, pp. 6–14; January, 1957.
[10] H. L. Armstrong, "On the switching transient in the forward conduction of semiconductor diodes," IRE Trans. on Electron Devices, vol. ED-4, pp. 111–113; April, 1957.
[11] C. D. Simmons, private communication, Applications Dept., Lansdale Tube Co., Division of Philco Corp., Lansdale, Pa.
[12] R. L. Pritchard, "Frequency variation of junction transistor parameters," Proc. IRE, vol., 42, pp. 786–799; May, 1954.
[13] R. A. Henle and J. L. Walsh, "The application of transistors to computers," Proc. IRE, vol. 46, pp. 1240–1254; June, 1958.
[14] L. P. Hunter, "Handbook of Semiconductor Electronics," McGraw-Hill Book Co., Inc., New York, N. Y.; 1957.
[15] T. R. Bashkow, "Effect of nonlinear collector capacitance on collector current rise time," IRE Trans. on Electron Devices, vol. ED-3, pp. 167–172; October, 1956.
[16] H. S. Yourke and E. J. Slobodzinski, "Millisecond transistors current switching techniques," *Proc. WJCC*, pp. 68–72; February, 1957.

Comparative Performance of Saturating and Current-Clamped High-Frequency Pulse Circuits*

V. P. MATHIS†, senior member, ire, H. RAILLARD†, member, ire, and J. J. SURAN†, senior member, ire

THE relative performance of saturating and current-clamped flip-flops are compared with regard to such properties as stability, loading capability, lse repetition frequency, trigger energy requirement, d pulse propagation time. It is shown both theoretically and experimentally that the two-terminal driving-int resistance characteristics of the saturating and rrent-clamped circuits are virtually identical and nce the static properties, *e.g.*, stability and loading pabilities, are the same for most practical purposes. ne saturating circuit has a small advantage insofar as collector-to-emitter voltage drop, when conducting, approximately one-half volt less than for the clamped rcuit, and consequently the saturating circuit may be aded a little more.

The clamped circuit is advantageous in dynamic performance only when the minority-carrier storage factor the clamping diode is less than the storage factor of the transistor. Using clamping diodes of the S347G and S555G types, the clamped circuit was definitely superior in dynamic performance to the saturating circuit when transistors of the 2N428 class were used. However, the clamped circuit showed little or no advantages when the 2N501 transistor was employed. The average experimental results obtained for the logic driving flip-flops illustrated are summarized in Table I.

TABLE I

Dynamic Circuit Property	2N428		2N501	
	Saturating	Clamped	Saturating	Clamped
Trigger Charge (coulombs)	30×10^{-10}	7×10^{-10}	2.5×10^{-10}	1.8×10^{-10}
Propagation Time (mμsec)	170	100	12	17
Repetition Rate (5-volt trigger)	350 kc†	1 mc*	1.7 mc*	1.7 mc*

* Limited by flip-flop coupling-network time constant.
† Limited by trigger-network time constant.

* Abstract of paper appearing in *Semiconductor Products*, vol. 3, bruary, 1960. This paper was presented at the Sixth Annual Den- Research Institute Symposium on Computers and Data Proc-ing, Estes Park, Colo., July, 1959.

† General Electric Co., Syracuse, N. Y.

A New Core Switch for Magnetic Matrix Stores and Other Purposes*

I. P. V. CARTER†

Summary—This paper analyzes the conventional uses of magnetic switch cores to drive matrix stores in both current-driven and voltage-driven modes. A new method of using switch cores is proposed and analyzed which offers, at the cost of replacing in every selection line the usual switch-core and terminating resistor by two smaller cores, intrinsic pulse shaping and amplitude regulation, and much reduced power dissipation, particularly in the driving stages. Constructional details of an application of the new method to drive a store 100×80×10 are given, and waveforms for this store are shown. All address decoding and driving are performed by 34 transistors. A model of a multiple coincidence store 101×101 with a cycle time of 1 μsec has also been constructed; details are given.

I. Introduction

THE principles of magnetic core matrix storage are now well known:[1–16] a brief account will be found in Appendix I. In order to drive a conventional double-coincidence storage matrix, a symmetrical current output must be produced of the form shown in Fig. 1, where the pulse amplitude I_2, length T and rise time τ are determined by the properties of the storag cores, while the decay time τ' and the interval betwee pulses (and hence the cycle time) are usually require to be as short as possible.

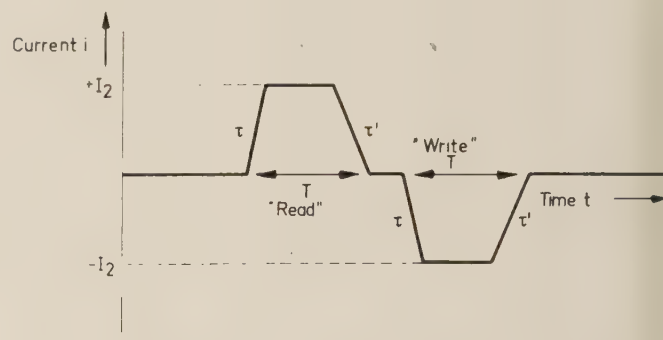


Fig. 1—Selection-line current in a typical storage matrix.

Magnetic cores with rectangular hysteresis loops hav been used as switches[8–15] to perform the last stage c address decoding and at the same time act as trans formers driving the storage matrix selection lines. Thes cores are here called switch cores (to distinguish thei from storage or memory cores), and are commonly as sembled in switch matrices.

While a great deal of information has been publishe on the organization and general working of suc switches,[5–15] relatively little (with the exception c Renwick's paper[8]) has appeared concerning the actua design methods used and the relative efficiencie achieved. Therefore, it is the purpose of the first part c this paper to analyze the operation of the conventiona core switch system in some detail.

The simplifying assumption is made that the loa does not display large transmission line effects, which justified if the propagation time down the selection lin and back again is less than the rise time of the curren pulse.

A small part of the flux from the switch core is use to establish the required current in the load inductanc L, and the remaining flux is dissipated during the pulse compensating the voltage drop across the total resist ance R around the selection loop. When the switch cor saturates, it acts as an effective short-circuit and th load current decays with the time constant L/R. T make this decay reasonably fast, the line resistance and, consequently, the flux linkage of the switch cor have to be made relatively large; and hence the tota energy input per pulse also becomes very large com pared with the energy actually stored in the inductiv part of the load. Also, since the switch core acts as

* Received by the PGEC, November 5, 1959; revised manuscript received, February 25, 1960.

† Research Laboratory Zurich, IBM Corp., Adliswil-Zurich, Switzerland.

[1] J. A. Rajchman, "A myriabit magnetic-core matrix memory," Proc. IRE, vol. 41, pp. 1407–1421; October, 1953.

[2] A. A. Robinson, V. L. Newhouse, M. J. Freidman, D. G. Bindon, and I. P. V. Carter, "A digital store using a magnetic core matrix," *Proc. IEE (London)*, vol. 103, pt. B, suppl. no. 2, pp. 295–301; April, 1956.

[3] C. J. Quartly, A. L. Cain, and R. W. Clarke, "A matrix store for data rate conversion," *Proc. IEE (London)*, vol. 104, pt. B, suppl. no. 7, pp. 457–463; November, 1956.

[4] W. N. Papian, "High speed computer stores 2.5 megabits," *Electronics*, vol. 30, pp. 162–167; October, 1957.

[5] R. L. Best, "Memory units in the Lincoln TX-2," *Proc. WJCC*, pp. 160–167; February, 1957.

[6] R. K. Richards, "Digital Computer Components and Circuits," D. Van Nostrand Co., Inc., New York, N. Y., ch. 8; 1957.

[7] J. A. Rajchman, "A survey of magnetic and other solid state devices for the manipulation of information," IRE Trans. on Circuit Theory, vol. CT-4, pp. 210–225; September, 1957.

[8] W. Renwick, "A magnetic core matrix store with direct selection using a magnetic core switch matrix," *Proc. IEE (London)*, vol. 104, pt. B, suppl. no. 7, pp. 436–444; November, 1956.

[9] F. H. Wells and J. G. Page, "A 1,000 channel neutron-velocity spectrometer, using ferrite data storage," *Proc. IEE (London)*, vol. 104, pt. B, suppl. no. 7, pp. 470–480; November, 1956.

[10] E. Foss and R. S. Partridge, "A 32,000 word magnetic-core memory," *IBM J. Res. and Dev.*, vol. 1, pp. 102–109; April, 1957.

[11] R. E. Merwin, "The IBM 705 EDPM memory system," IRE Trans. on Electronic Computers, vol. EC-5, pp. 219–223; December, 1956.

[12] J. L. Mitchell and K. H. Olsen, "TX-O, a transistor computer with a 256×256 memory," *Proc. EJCC*, pp. 93–101; December, 1956.

[13] D. W. Willis, "A ferrite-core switch-matrix for magnetic-recording-head selection," *Proc. IEE (London)*, vol. 104, pt. B, suppl. no. 7, pp. 464–469; November, 1956.

[14] H. Gillert, "Bemessung der Bauelemente eines Schnellspeichers mit Ferrit-Ringkernen," *Nachrtech. Z.*, vol. 9, pp. 250–252; June, 1956. (In German.)

[15] E. L. Younker, "A transistor-driven magnetic core memory," IRE Trans. on Electronic Computers, vol. EC-6, pp. 14–20; March, 1957.

[16] R. C. Minnick and R. L. Ashenhurst, "Multiple coincidence magnetic storage systems," *J. Appl. Phys.*, vol. 26, pp. 575–579; May, 1955.

nsformer throughout the pulse, any variation of mary current or voltage is transmitted to the load. ese processes are analyzed in Section III of this paper. A new way of using switch cores which overcomes se problems is described and analyzed in Section IV. r the same value of load inductance as before, the d resistance is reduced to a minimum, and the switch- e flux linkage is reduced to a value just sufficient to ablish the required current in the load. Hence, the tch core saturates at the rise of the primary pulse, l subsequent variations at the primary no longer ect the output. The secondary current is now deter- ned only by the ratio of the flux linkage in the switch e to the inductance of the load.

The top of the pulse so formed is almost flat, since the ue of L/R has deliberately been made large. The se is terminated by a second switch core operating the same way in opposition to the first one. This angement will be referred to as a two-core switch and n be driven very hard to achieve high speeds without ecting current regulation; it also has the interesting property that the write pulse (Fig. 1), produced by consecutively resetting the two cores, can be suppressed completely by simultaneous resetting of both cores.

This circuit is analyzed in Section IV, and some details of practical designs are discussed in Section V. The two cases are critically compared in Section VI, and some applications with experimental results are described in Section VII. The main conclusions which can be drawn from the work presented are given in Section VIII.

II. Symbols

All symbols are defined where they first appear. The following list is merely a summary. All quantities varying with time are denoted by small letters; most constants are denoted by capitals or by Greek letters. Suffixes 1 and 2 denote primary and secondary, respectively, except where they refer to the two cores of Figs. 7, 16 and 18; suffixes A and B refer to the circuits of Figs. 10 and 11, respectively; other suffixes are self-explanatory. MKS units are used throughout.

L = Load inductance
T = Pulse length
τ = Pulse rise time (0–90 per cent)
τ' = Pulse decay time (100 per cent–10 per cent)
I_2 = Pulse amplitude
} design data

R = Load resistance (also R_A, R_B)
C_B = See Fig. 11
J_0 = See Appendix IV
S = Core switching coefficient
} core material properties

t_s = Core switching time
J_c = Coercive current
$\phi(t)$ = Flux state of core
Φ = Irreversible flux available (also Φ_A, Φ_B)
k = Reversible inductance
l = Magnetic path length
} core properties: see Fig. 2

n = Number of turns (also n_1, n_2, n_A, n_B)
L' = Stray inductance of $(m-1)$ cores (also L_A', L_B')
$L^* = L + n_1^2 k + n_2^2 k$ (see Appendix II)
$i_1(t)$ = Primary current (pulse amplitude I_1)
$i_2(t)$ = Secondary current (pulse amplitude I_2)
I_3 = Current pulse amplitude (Fig. 7)
I_m = Maximum current from assumed voltage source
$i_R(t)$ = Current in R_B (Fig. 21)
$i_L(t)$ = Current in L (Fig. 21)
$u_1 u_2 v_1 v_2$ = See Fig. 16
} currents

$e(t)$ = Voltage across primary winding (pulse amplitude E)
E_A, E_B = Collector supply voltages (Figs. 10 and 11)
$v(t)$ = Voltage across R_B (Fig. 21)
} voltages

t = Time
t_1 = Time (arbitrarily small)
$\gamma = 1 + 2.3\ T/\tau$ [Section VI, (20)]
δ = Flux loss in load during pulse (see Appendix II)
m = Number of cores in collector circuit (Figs. 10 and 11)
W_c = Energy dissipated per pulse in core
W_T = Energy dissipated per pulse in transistor.

III. Analysis of the Conventional Switch

This section considers the behavior of a core with an almost rectangular hysteresis loop, which is loaded by an inductance L in series with a resistance R. The assumed hysteresis loop is shown in Fig. 2: it has vertical sides, the coercive current is J_c ampere-turns, the irreversible flux per turn available is Φ webers per turn, and the slope of the top and bottom sections of the loop (assumed to be straight lines) is k webers per turn per ampere-turn (1 weber = 1 volt-second; 1 weber per ampere = 1 henry). Both primary and secondary windings are assumed to have n turns—this involves no loss of generality and simplifies the mathematics. Fig. 3 shows the characteristics of this winding.

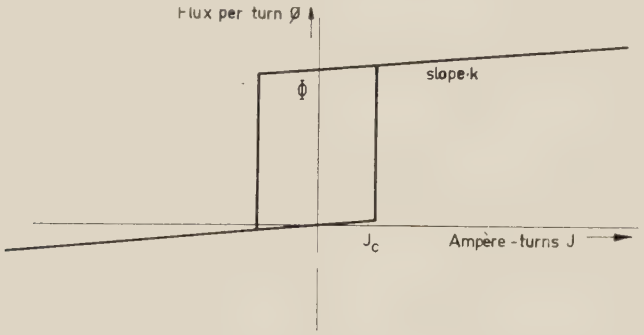


Fig. 2—Idealized hysteresis loop of a switch core.

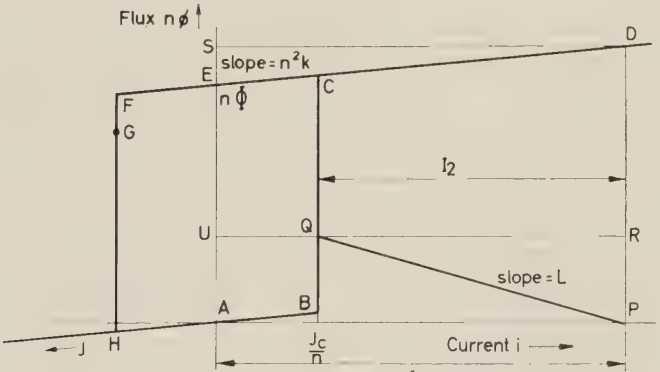


Fig. 3—Flux-current diagram for a loaded switch core.

The primary current flowing will be denoted by $i_1 = i_1(t)$; the secondary current by $i_2 = i_2(t)$. We shall take

$$i_1(t) = 0 \quad \text{for} \quad t < 0$$
$$= I_1 \quad \text{for} \quad t > 0.$$

The core is initially at state A with no current flowing in the secondary winding. If $nI_1 \leq J_c$, the core behaves linearly and remains on the line between A and B (Fig. 3); this case is of no interest. If $nI_1 > J_c$, a current i_2 will flow in the secondary opposing I_1. An emf $e(t)$ will appear across both primary and secondary windings such that

$$e(t) = n\frac{d\phi}{dt} = i_2R + L\frac{di_2}{dt} \qquad (1)$$

(neglecting negative signs for simplicity). Integrating we have

$$\int_0^t e dt = n\phi(t) = R\int_0^t i_2 dt + Li_2(t). \qquad (2)$$

Consider the state of affairs immediately after $t = 0$, $t = t_1$. Since t_1 can be taken arbitrarily small and since i_2 is always finite, the first term of (2) can be neglected. Hence, $n\phi(t_1) = Li_2(t_1)$, which means that the state of the core will move during the time t_1 to a point Q on the hysteresis loop such that the flux change it undergoes is equal to that undergone by the load inductance. $i_2(t_1)$ will then be the difference between the primary current I_1 and the current J_c/n required to link the core at Q. Point Q will be between A and B if

$$I_1 < \frac{J_c}{n}\left(1 + \frac{n^2k}{L}\right),$$

between B and C if

$$\frac{J_c}{n}\left(1 + \frac{n^2k}{L}\right) < I_1 < \frac{J_c}{n}\left(1 + \frac{n^2k}{L}\right) + \frac{n\Phi}{L},$$

and between C and D if

$$I_1 > \frac{J_c}{n}\left(1 + \frac{n^2k}{L}\right) + \frac{n\Phi}{L}.$$

The case where Q falls between A and B is not interesting and rarely occurs in practice; it will not be considered any further. The case where Q falls between C and D will be considered in Section IV. All conventional uses of switch cores known to the author involve Q falling between B and C, and this case will be considered in the remainder of this section.

A geometrical construction for finding Q is shown in Fig. 3. P is the point $(I_1, 0)$. From P a line is drawn backwards of slope L to intersect the loop at Q; in the example shown, this point Q is between B and C. From Q, a horizontal line is drawn to intersect a vertical line through P at R: then, since the slope of PQ is equal to L, PR represents the flux in L when a current QR flows in it. Thus, $PR = n\phi(t_1)$, $OP = I_1$ and $QR = i_2(t_1)$.

In the conventional core-switch design, the output current is $I_2 = i_2(t_1)$ (see Fig. 1). Hence,

$$I_1 - I_2 = \frac{J_c}{n} \qquad (3)$$

in the example shown.

The effect of damping losses, *i.e.*, the finite switching speed of the core, will be considered below. At $t = t_1$ the core is at Q and a current I_2 flows in the secondary winding. This current remains constant for the next period of time, and (1) becomes

$$e = n\frac{d\phi}{dt} = RI_2. \tag{4}$$

e core moves from Q toward C at a constant rate de-mined by (4). Eq. (2) becomes

$$n\phi(t) = RI_2t + LI_2, \tag{5}$$

e i_2 is constant and equal to I_2.
he core reaches C after a time T given by

$$n\Phi + n^2k\cdot\frac{J_c}{n} = RTI_2 + LI_2,$$

$$T = (n\Phi + nkJ_c - LI_2)/RI_2.$$

neglect nkJ_c and write

$$T = \frac{n\Phi - LI_2}{RI_2} \text{ approximately.} \tag{6}$$

er this, the core moves from C towards D. We have, n (1),

$$n\frac{d\phi}{dt} = -n^2k\frac{di_2}{dt} = i_2R + L\frac{di_2}{dt}$$

nce,

$$(L + n^2k)\frac{di_2}{dt} + i_2R = 0, \tag{7}$$

solution of which describes the decay of the output rent:

$$i_2(t) = I_2\cdot\exp\left\{\frac{(T-t)\cdot R}{L + n^2k}\right\} \tag{8}$$

id for $t>T$ when the known initial conditions are erted. From this the decay time τ' is seen to be $(L+n^2k)/R$.

he physical interpretation of this process is as fol-s. Between $t=0$ and $t=t_1$, a finite quantity of flux $(t_1)=LI_2$ is transferred from the core to the induct-e L. For the rest of the time, flux is dissipated in the istance. This flux comes entirely from the core, until T, owing to the steepness of the sides of the hystere-loop; but afterwards it comes partly from the load uctance as well, thus reducing the secondary cur-t and, therefore, also the rate of flux dissipation. By end of the process, all the flux extracted from the e in going from A to D (*i.e.*, $n\Phi+n^2kI_1$) has been dis-ated in the resistance. Fig. 4 shows i_2 plotted as a ction of time. Fig. 5 is a typical photograph of an ual ferrite core working under these conditions.

he limiting factor in the design is the decay time of trailing edge of the pulse. Attempts to reduce this olve increasing R while T is kept constant; this re-res more flux from the core [see (5)]. In terms of Fig. we try to increase the ratio QC:QB; and since QB constant, depending only on the load and the required output current, we have to increase QC, either by increasing n or by taking a core with larger cross-sectional area. In both cases, improvement is at the cost of extra power and is partially offset by the increased slope of ED. This vicious circle is inescapable in the conventional case. It is, of course, always possible to terminate the first pulse (of Fig. 1) by terminating the primary current. This would normally be done when the core has just reached the point C [Fig. 5(b)], and the core will then move to some point G such that the flux change CG is equal to

$$\left(I_2 - \frac{J_c}{n}\right)L$$

and a small current J_c/n continues to flow in the sec-

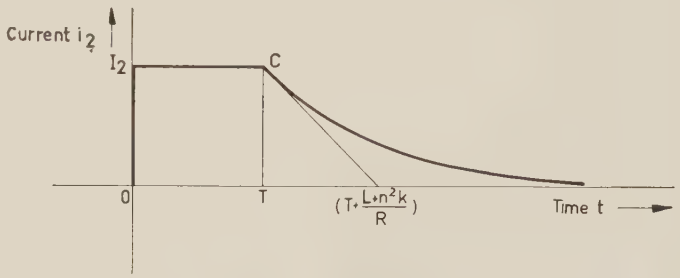
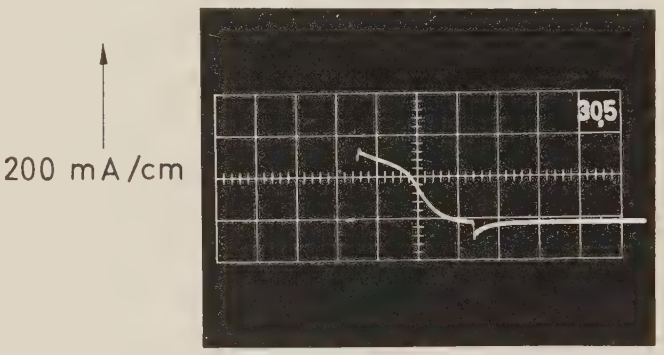
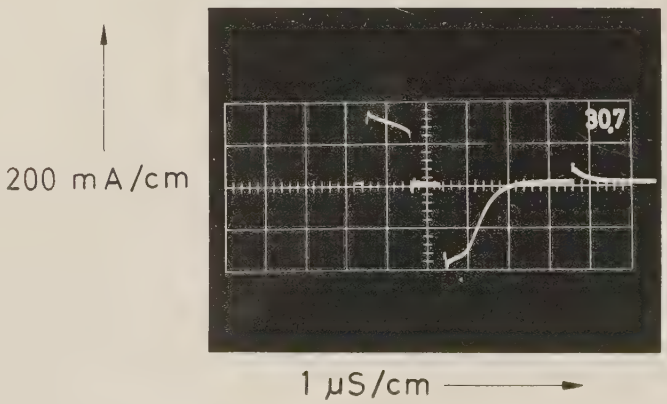


Fig. 4—Output from a switch core, conventionally operated.



(a)



(b)

Fig. 5—Outputs from a ferrite switch core, conventionally operated.

ondary. This means that the flux available for the second pulse is less than that available for the first one by this amount, and this must be allowed for in the design. This is done by noticing that, for the second pulse, the irreversible flux available is only HG instead of HF, where FG is approximately equal to QB. We are once more in the position of having to increase the height of the loop, with the attendant disadvantages of increased power consumption and this time (since we cannot increase R), increased decay time at the end of the second pulse. It should be noted that the second pulse usually cannot be terminated in the same way, because the core is usually also used for selection, and must, therefore, be left either at J or at A. However, it may happen that there is more time available for pulse decay after the second pulse than after the first, in which case this technique may be used.

IV. The Two-Core Switch

As an introduction to the two-core switch system we consider the case so far postponed where the point Q of Fig. 3 falls on CD (see Fig. 7), *i.e.*, where

$$I_1 > \frac{J_c}{n}\left(1 + \frac{n^2k}{L}\right) + \frac{n\Phi}{L}.$$

We now have a new value of $i_2(t)$ which, as before, must be equal to the design figure I_2. From (2), as previously, we have

$$n\phi(t_1) = LI_2$$

from which

$$LI_2 = n\Phi + n^2k(I_1 - I_2),$$

i.e.,

$$I_2(L + n^2k) = n\Phi + I_1 \cdot n^2k$$

and

$$I_2 = \frac{n\Phi}{L + n^2k} + \frac{I_1 n^2k}{L + n^2k}. \qquad (9)$$

Subsequently, the core moves from Q towards D exactly as before [(7) still applies, with new initial conditions], and the complete solution is

$$i_2(t) = I_2 \cdot \exp - \left(\frac{Rt}{L + n^2k}\right)$$
$$= \frac{n\Phi + n^2kI_1}{L + n^2k} \cdot \exp - \left(\frac{Rt}{L + n^2k}\right) \qquad (10)$$

valid for $t > t_1$. This is illustrated in Fig. 6.

It is now necessary to make the decay time constant $(L+n^2k)/R$ large compared with T, and the added resistors of the conventional system are no longer required.

The possibility of terminating the output pulse by terminating I_1 exists as before. If this is done in a time which is short compared with the decay time of I_2, then the core will almost completely be reversed in t process; from the physical point of view, the fl switched initially from the core to the load has almo all been reabsorbed, apart from some losses caused b resistance or storage-cores in the load and a small cu rent (equal to J_c/n) left circulating in the secondar This, of course, makes it virtually impossible to obta a useful negative output pulse; on the contrary, the conditions are clearly useful only when no negative ou put is required, which state of affairs is easily realize provided the load losses are kept to a minimum (entai ing, among other things, a small value of R). The cu rent waveform obtained is, of course, still subject to th requirement that the area above zero is equal to tha below zero, which must be considered during design.

A much more flexible arrangement is to termina the pulse by switching a second core, whose secondar winding is in series with the first; the two cores can b identical. This arrangement is the basis of the "tw core switch," and a wiring diagram of the basic patter is shown in Fig. 7. The convention used in this diagra is that of Karnaugh,[17] where cores are represented b vertical lines, windings by horizontal lines, and th sense of the winding by short diagonal lines ("mirr symbols") at their intersections. The direction of th MMF resulting from a current is obtained by reflectio

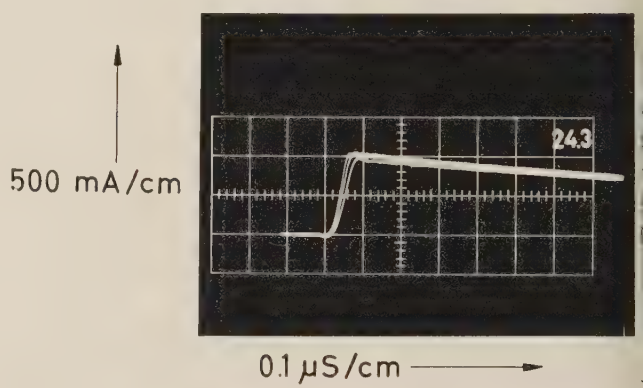


Fig. 6—Outputs from a switch core with an inductive load and inputs of 8, 10 and 12 ampere turns.

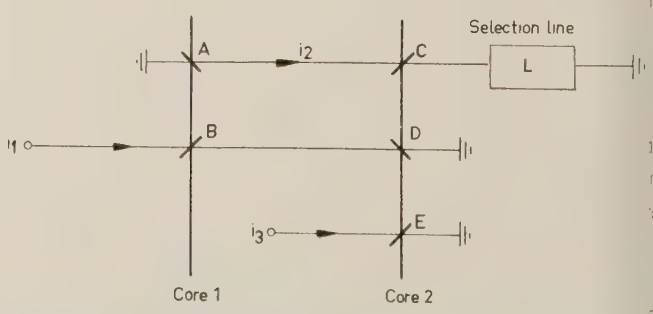


Fig. 7—Circuit of the two-core switch. Windings A and B have n_1 turns. Windings C, D and E have n_1 turns.

[17] M. Karnaugh, "Pulse switching circuits using magnetic cores," Proc. IRE, vol. 43, pp. 570–584; May, 1955.

he current through 90° by the mirror; the direction the EMF resulting from a flux change is similarly ained by reflection of the negative of the flux change ough 90°.

he operation of this circuit is as follows. To begin h, both cores are at A (Fig. 8). At $t=0$, I_1 is applied windings B and D (Fig. 7), thus switching core 1 and ving core 2 negative (toward J in Fig. 8). A current s established in the load L exactly as before; if we e $k=0$ as a rather crude approximation, then we ve, from (9),

$$I_2 = \frac{n_1\Phi}{L}. \tag{11}$$

n exact calculation is performed in Appendix II.) nilarly [cf. (10)] we have

$$i_2(t) = \frac{n_1\Phi}{L}\cdot e^{-(R/L)t}. \tag{12}$$

this approximation, the value of I_1 is immaterial, vided that

$$I_1 > \frac{J_c}{n_1} + \frac{n_1\Phi}{L}. \tag{13}$$

is condition insures that Q falls on CD (Fig. 8).

t will be noticed that the purpose of winding D (Fig. is to prevent core 2 from being switched by I_2.

At time $t=T$, a step function of current I_3 is applied to nding E, and switches core 2. This causes a current nagnitude $n_2\Phi/L$ to flow in winding C, opposing the rent already flowing there. This current will have de- ed slightly, according to (12), from its initial value; he load is nonlinear, it may be less than that indicated (12). In either case, n_2 is chosen such that

$$\frac{n_2\Phi}{L} = i_2(T) \tag{14}$$

ere $i_2(T)$ is the value of i_2 just before the second core witched. This will insure that the current in L is re- ed exactly to zero when core 2 is switched.

n order to insure the complete switching of core 2, nust be large enough; we must have

$$I_3 > \frac{J_c}{n_2} + I_1. \tag{15}$$

er this, the currents I_1 and I_3 can be switched off, that order, no output currents occurring in L. This ves both cores at E (Fig. 8).

A negative output pulse can then be made in the ne way. However, it is also possible to restore both es to A simultaneously; this will cause a current $-n_2)\Phi/L$ to flow in the secondary. In order to make s current as small as possible, it is necessary to make n_2 rly equal to n_1, and, therefore, the losses during the itive pulse must be kept small. This means [from (12)] that L/R should be large compared with T. The output of such a switch is shown in Fig. 9.

So far in this discussion, we have taken $k=0$. The exact calculation for $k>0$ is given in Appendix II; here we note simply that the effect of a nonzero but small value of k (*i.e.*, $n^2k \ll L$) is to make the initial value of i_2 slightly larger and slightly dependent on I_1 [as in (9)], and to cause small currents to flow when I_1 and I_3 are switched off. It may be desirable to choose n_2 differently, according to the particular time during the cycle that these out-of-balance currents are required to be a minimum, and according to the various combinations of load and cycle likely to be encountered. In a matrix store, for example, a storage core may or may not be switched during either pulse, and the second pulse may or may not be present. Different combinations of these possibilities result in different out-of-balance currents, and hence a careful choice of n_2 is necessary. These out-of-balance currents decay with a time constant $(L+kn_1^2+kn_2^2)/R$. Therefore, this should, if possible, not be large compared with a complete cycle time, consistent with its being large compared with T.

V. Behavior of Switches in Practice

In practice, our assumptions are never exactly fulfilled, and a number of minor effects must therefore be expected. These are discussed in the following paragraphs.

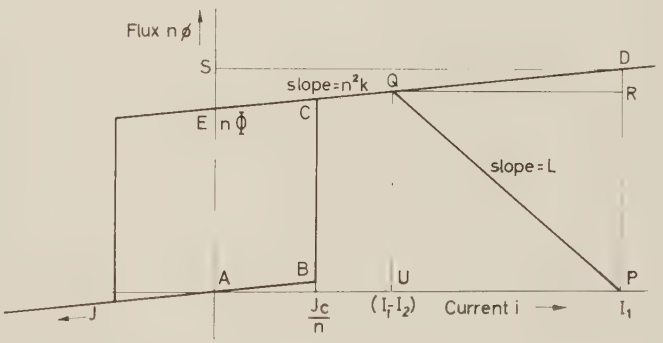


Fig. 8—Flux-current diagram for a loaded switch core.

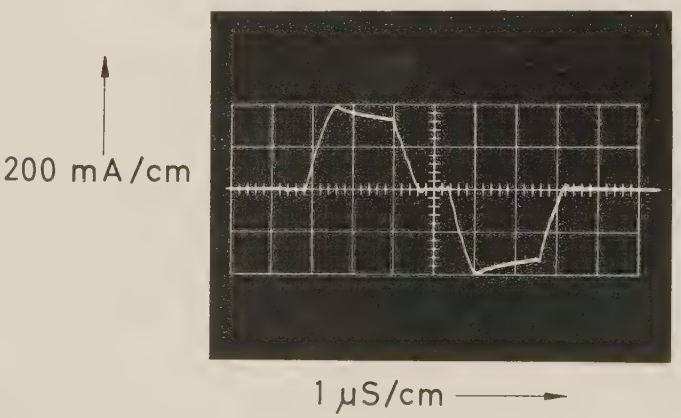


Fig. 9—Output from a two-core switch driving a load of 63 μh in series with 7 ohms.

A. Regulation

In the conventional switch, the core behaves as a transformer throughout the output pulse, and variations in primary current are transmitted to the output. This is readily seen from (3); I_2 is directly proportional to I_1. In addition, T will vary inversely as I_1. Most published designs[9–13] use some form of feedback amplifier to stabilize I_1 and/or I_2.

In the two-core switch, I_2 is almost independent of I_1, since the core is saturated during the current pulse and the coupling between primary and secondary is small. No stabilization of I_1 is necessary. [See (10) and Fig. 6.]

However, the output current is now directly proportional to the switch-core flux, and cores must therefore be selected on installation whose remanence lies within the necessary limits; existing core handling and testing equipment should be adequate for this. The remanence of ferrite cores decreases with temperature; this effect will automatically compensate for the corresponding decrease of coercive current in the storage cores, provided the two temperature coefficients are matched. (But if such a technique is to be successful, the power dissipation in the switch cores must be kept small.)

B. Finite Rise Time

For both conventional and two-core switches, a slow rise of I_1 will result in a slow rise of I_2.

C. Core Switching Time

In the conventional switch, the effect of the secondary circuit is to constrain the core, and to force it to switch more slowly than it would if the secondary circuit were omitted. If T is large compared with this open-circuit core-switching time, then the analysis given above is adequately accurate, provided a dynamic hysteresis loop (*i.e.*, one plotted from the observed pulse response of the core) is used. Where a value of T approaching the open circuit switching time is required, allowance for damping losses can be made either by use of a larger effective value of J_c or by insertion of an effective shunt resistance. Neither method is accurate, and trial designs should always be verified experimentally.

In the two-core switch, the switching speed of the core no longer affects the pulse amplitude I_2. The pulse rise time τ, as before, depends on the core-switching speed but can be reduced if necessary by driving the core harder. Because of the intrinsic regulating properties of the two-core switch, this will not affect I_2.

D. Sloping Sides of the Loop

In the conventional switch, if the sides of the loop are not vertical but are still fairly steep, it is clear that as the core moves from Q to C (Fig. 3), i_2 will decrease. This effect is clearly shown in Fig. 5, appearing as a slight slope of the top of the pulse; it will become less important if nI_1 is made large compared with J_c. Th rounding of the corner of the loop at C will in a simil way cause the rounding of the corresponding part Fig. 5.

In the two-core switch the slope of the sides of th loop will, of course, have no effect on the output puls

E. Nonlinear Load

When the load consists of a storage-matrix selectio line, a core switching in the load will give the effect of nonlinearity. In practice, for matrices about 64×64 larger, this will be quite small compared with the oth impedance, and its effect in the conventional switch ca easily be seen from Fig. 3. If L now represents only th linear part of the load, and if the matrix core switches a time a little shorter than T (as will be the case in pra tice), the flux absorbed by the storage core will com from the switch core, and hence T will be smaller by corresponding amount. The decay time of the pulse wi be unaffected. The variation in T must, of course, b allowed for in the design, if it is not entirely negligibl

In the two-core switch, such nonlinearities, resultin from storage cores switching, will remove flux stored the load inductance and reduce I_2. This effect will b spread over the time T and will be added to the deca already present because of resistance. It must, ther fore, be considered during design. If a number of sto age planes are connected in series, the losses will deper on the number of cores switched (in reading or writing and, therefore, the effect will vary according to the co in which information is stored. In binary code, allow ance must be made for any number of cores switchin from zero to one per plane; but in certain codes (*e.g* two-out-of-five), the number is always the same (tw cores per five planes), and in other codes (*e.g.*, exces three, binary-coded-decimal), the number is limite (either one, two or three cores per four planes).

F. Transmission Line Effects

When several selection lines are connected in serie the resulting distributed self-capacitance can cau transmission line effects such as appreciable dela times and reflections. A full discussion of such effects beyond the scope of this paper. It can be said, howeve that as long as the delay time caused by such effects not greater than the required output-pulse rise time, th effect can be ignored. This will be the case for all but th largest stores.

Transmission line effects can be reduced by arrang ment of the geometry of the array so as to minimi stray capacitance. In the conventional switch, deleter ous effects caused by reflections can be minimized b arranging that R is equal to the effective line impedanc this solution is not possible with the two-core switch.

G. Stray Capacitance

The effect of stray capacitance is to cause slower ri times, overshoot and ringing. The effect is much le

arked in the two-core switch because of its inherent ipping effect which isolates the secondary from ring- g on the primary side. This clipping is clearly seen in ig. 9.

. *Efficiency*

The efficiency of the conventional switch is perhaps est seen from Fig. 3. The total energy input during a ulse is equal to the area $APDS$; the energy stored in ie inductive part of the load is equal to the area PRQ. his is a fair comparison since, when the load is a stor- ge matrix selection line, the actual load resistance ould be extremely small. A resistor must be inserted series with every selection line in order to achieve the ecessary decay time at the end of the pulse ($\tau' = 2 \cdot 3(L + n^2k)/R$). Of the total input energy, an amount $PQCD$ dissipated in the load resistance, $APQCE$ is dissi- ated in the core, EDS is recovered from the core, and ssipated in the load at the end of the pulse. The total put energy in the two-core switch is given by the area PDS in Fig. 8. Of this, PQR is stored in the load in- uctance. This is clearly a very much bigger fraction of ie total than in the previous case (Fig. 3). $APQE$ is ssipated in the core, and EDS is recovered from the ore and dissipated in the load resistance at the end of ie pulse.

VI. A Comparison

The analysis of the previous section dealt with cur- nt-driven switches, since this is at present standard ractice in most core stores. The two-core switch, how- ver, offers very substantial power saving which permits ie use of transistor drivers operating in the voltage ode. We shall suppose, therefore, that a given store is be driven by a switch-core matrix driven by tran- stors operated as voltage sources. The required selec- on-line waveform is shown in Fig. 1 and, as before, is pecified by the four parameters I_2, T, τ and τ'. The ad has inductance L as before. The store is of the con- entional double-coincidence type.

The circuits to be compared are shown in Figs. 10 nd 11. The conventional system is called A and the wo-core system is called B. In practice, as shown, a ngle transistor will have in its collector circuit a num- er of switch cores connected in series, but all except ne of these will be prevented from switching by suitable ias currents (not shown). For the moment, their im- edance will be neglected. The switched core will be onsidered to have a turns ratio of 1:1 in each case, and load of inductance L (simulating a selection line). In ase A, as we have seen, it is standard practice to add a esistor R_A in series with L. This serves the dual purpose achieving the necessary rise and fall times of the ulse and of stabilizing the output current by swamping ie effect of variations in the number of storage cores witched. In case B, in order to prevent excessive power issipation in the transistor, a resistor R_B is added as

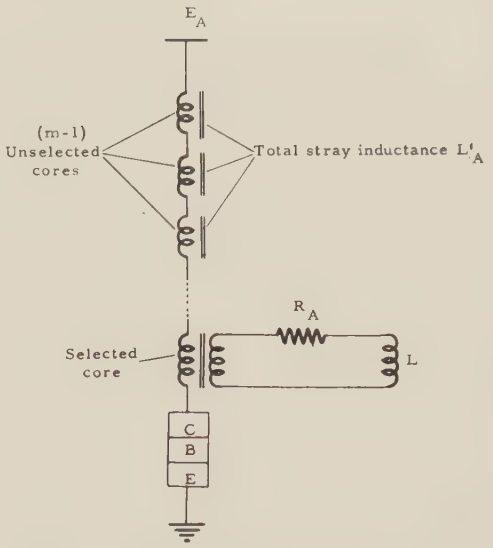


Fig. 10—Conventional core drive (Case A).

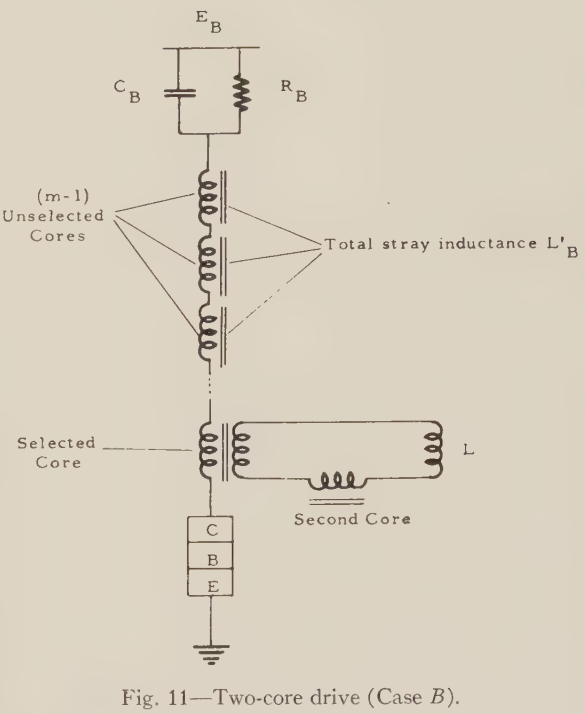


Fig. 11—Two-core drive (Case B).

shown, in series with the collector. This is decoupled with a capacitor C_B. It should be noted that a resistor R_A is required for every selection line L, but the pair R_BC_B will serve a complete set of transistors from which only one is to be selected at a time. Hence, only four pairs R_BC_B are required for a complete store. The collector supply voltages are E_A and E_B.

The values of R_A, R_B, C_B, E_A and E_B, as well as the switch-core data, can be calculated as follows.

Case A

A detailed calculation is given in Appendix III. It is shown there that the output current rises exponentially, with a rise time (0–90 per cent), as follows [from (42)]:

$$\tau = 2{\cdot}3\frac{L}{R_A} = \tau'. \tag{16}$$

Hence,

$$R_A = \frac{2.3L}{\tau} \tag{17}$$

and

$$E_A = I_2R_A = 2.3\ I_2L/\tau \tag{18}$$

$$n_A\Phi_A = I_2R_AT + I_2L = I_2L\left(1 + 2.3\frac{T}{\tau}\right) \tag{19}$$

$$= I_2L\gamma \quad \text{where} \quad \gamma = 1 + 2.3\frac{T}{\tau}\cdot \tag{20}$$

Case B

When the voltage step is applied, the core will switch completely during the rise of the secondary-current pulse, and will subsequently present a short circuit to the source. We have, therefore,

$$\tau = \frac{n\Phi}{E_B} = \frac{LI_2}{E_B} \tag{21}$$

and, hence,

$$E_B = \frac{L\cdot I_2}{\tau} \tag{22}$$

$$R_B = 0.75\frac{E_B}{I_2} \tag{23}$$

$$n_B\Phi_B = I_2L\tau \tag{24}$$

and

$$C_B = \frac{\tau}{R_B}\cdot \tag{25}$$

(See, also, Appendix IV.)

It is apparent, from (18) and (22), that $E_A = 2.3\ E_B$. This result is due to the possibility, in case B, of shunting R_B with C_B; this maintains almost the full collector supply voltage across L until the current pulse is established. Afterwards, the voltage across C_B builds up rapidly until the transistor saturates, thus avoiding excessive power dissipation in the transistor. At the end of the pulse, the network R_BC_B recovers (95 per cent) in a time 3τ. A full calculation is made in Appendix IV.

We shall assume that a complete optimum design for the core and winding in case A has been made, and a core material selected, core dimensions calculated, and n_A fixed. For the sake of argument, we shall retain the same core material, the same magnetic path length l, and the same value of n, for case B. We then have

$$n_A = n_B = n \tag{26}$$

and, from (20) and (24)

$$\Phi_B = \frac{\Phi_A}{\gamma} \tag{27}$$

$$k_B = \frac{k_A}{\gamma}\cdot \tag{28}$$

So far, we have neglected the stray inductance, n^2k. We can estimate its value in case A from Fig. 3; if we assume the squareness ratio (B_r/B_s) of the material to be say 0.98, then

$$n^2k_A = \frac{n\Phi_A}{50}\cdot\frac{n}{2J_c}\cdot$$

Now

$$L = \frac{n\Phi_A}{\gamma I_2}$$

and if we make $I_2 = 2J_c/$ as n and $\gamma = 10$ (about the smallest values practicable), we obtain

$$n^2k_A = \frac{L}{5}\cdot$$

Now the total stray inductance in the collector circuit of Fig. 10 is

$$L_A' = (m-1)n^2k_A \tag{29}$$

and this will be larger than L for $m > 6$. We also have from (28),

$$L_B' = \frac{L_A'}{\gamma}\cdot \tag{30}$$

We shall neglect the small difference (approximately J_c/n) between primary and secondary currents, and rewrite (18) and (22) as follows:

$$E_A = 2.3\frac{I_2}{\tau}(L + L_A') \tag{31}$$

$$E_B = \frac{I_2}{\tau}(L + L_B') \tag{32}$$

and since certainly $L_A' > L$ and $\gamma > 10$,

$$E_B < E_A/4. \tag{33}$$

This result is most important, as it means that the collector supply voltage in case B is only a quarter that of case A. In practice, this could mean that, with available transistors, a case B solution is possible where a case A solution is not; it may also be noted that by changing the turns ratio on the core, if the supply voltage is kept the same, only a quarter of the current is now required from the transistor. The energy W_c dissipated in the core during a single pulse is calculated in Appendix V. It is there found to be approximately equal to [from (61)]

$$W_c = J_0 \Phi l + \frac{lSE}{0.9n} \cdot \quad (34)$$

here S and J_0 are constants of the core material. Comaring cases A and B, we find the first term of (34) naller in case B in the ratio $(\Phi_A/\Phi_B) = \gamma$, say 10, and ie second term smaller in the ratio E_A/E_B, say 4. So ie energy dissipated in a switch core in case B is beween a quarter and a tenth of that dissipated in a witch core in case A.

The peak collector-emitter voltage applied to the ansistor is in each case $2E$ (applied whenever a core is versed in the collector circuit by another winding); us, in case A a transistor receives four times the peak oltage received by a transistor in case B, and this peak sts T/τ times longer. The power dissipation in the ansistors can be neglected (or at any rate is the same r both cases) for that part of the pulse during which e transistor is saturated (bottomed)—*i.e.*, with almost ro collector voltage. In case A, significant power dissiation can occur at the beginning of the pulse (because stray collector capacitance, the collector voltage does ot at once become zero) and at the end of the pulse. hen the core saturates, R_A no longer acts as a current miter, and the collector current rises to a limit I_m set the transistor characteristics and the preceding cirit. When this limit is reached, the supply voltage apars at the collector; but stray inductance prevents e collector voltage from rising until this limit is ached, and stray capacitance limits its subsequent te of rise. In theory, the transistor can be switched off a time τ' after the core saturates (the load current dys during this time), but in practice, some safety ctor is necessary. If we neglect stray capacitance, hose effects at start and finish cancel out (roughly), ll voltage will be applied to the transistor for a time least equal to τ' at the end of the pulse; the energy ssipated per pulse is then equal to $\tau' I_m E_A$. It may be oted that this dissipation can be avoided in the "read" ive transistors, by switching them off just before the re saturates, but, for reasons discussed in Section III, e write transistors cannot be turned off in this way. the write pulse is made by a standing dc bias, then e read transistors cannot be turned off prematurely, nce a gap must be left between read and write pulses allow the inhibit current in the storage matrix time become established.

In case B, dissipation also occurs both at the benning of the pulse and again when the core saturates. ne effect of stray capacitance can be neglected, as bere. The energy dissipation is calculated in Appendix and is equal to $\frac{1}{4}\tau I_m E_B$ (when $I_m = 2I_2$), *i.e.*, less than of that of case A (since $E_A > 4E_B$). The effect of ray inductance is to reduce this figure still further, as fore. Allowing for some inaccuracies in our assumpons, it is clear that in case B, transistor dissipation will rtainly be less than a quarter that of case A.

In a complete system, of course, there will be twice as many switch cores in case A as in case B; however, since the collector stray inductance in case B is so much less than that in case A, a single transistor can have many more cores in its collector circuit; and, hence, a typical store driven by two-core switches will contain fewer driving transistors than if it had been driven by conventional switches.

VII. Applications of the Two-Core Switch

A. Driving a Conventional Double Coincidence Matrix Store

The object here was to make use of the power-saving and self-regulating properties of the two-core switch in a coincident current memory. As a convenient available model, a storage array was chosen which consisted of ten planes, each 80×100. The measured impedance of a selection line (all ten planes connected in series) was 7.2 μh in series with 2.5 ohms for an X line (800 cores) and 9 μh in series with 3.5 ohms for a Y line (1000 cores). This gave an intrinsic value of L/R of about 2.5 μsec, which was too short to yield a sufficiently flat-topped pulse. For a new array, this state of affairs would readily be avoided by use of thicker wire (0.2 mm instead of 0.1 mm, and even thicker wire is possible) in the selection lines; but in this case this was not possible, so an external inductor was added in series with each line to increase the total inductance to about 22 μh. This was then equivalent to a store containing about 25 planes constructed with thicker wire.

The constructional details of the two-core switch development will be found in Appendix VI. The switch-core pairs are arranged in two matrices, 8×10 and $8 \times 12\frac{1}{2}$, to drive the X and Y selection lines, respectively. Address selection is performed by bias windings on each core pair, so arranged that the selected core pair is unbiased (none of its bias windings are energized) while all other core pairs are biased by one or more of their windings. Two drive windings, one for each core of a core pair, pass through all core pairs in both matrices, but since all core pairs except the two selected (one in each switch matrix) are prevented from switching by their bias, one X and one Y selection wire are simultaneously driven.[18]

The complete store contains four driving transistors and 30 bias transistors (the address was binary-coded decimal; in binary code, only 28 bias transistors would be necessary), with each address bit and its inverse driving one bias transistor.

The driving current waveforms are shown in Figs. 12 and 13, and the resulting selection line waveform in Fig. 14. Fig. 15 shows the output from the storage-matrix sense line (via a screened transformer). The store has a cycle time of 9 μsec.

[18] See, for example, Richards, *op. cit.*, Figs. 8–13a.

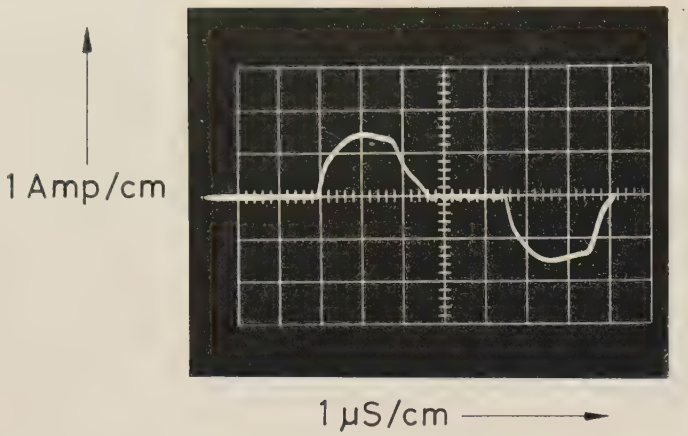


Fig. 12—Drive current for the first core in a two-core switch for a double-coincidence store.

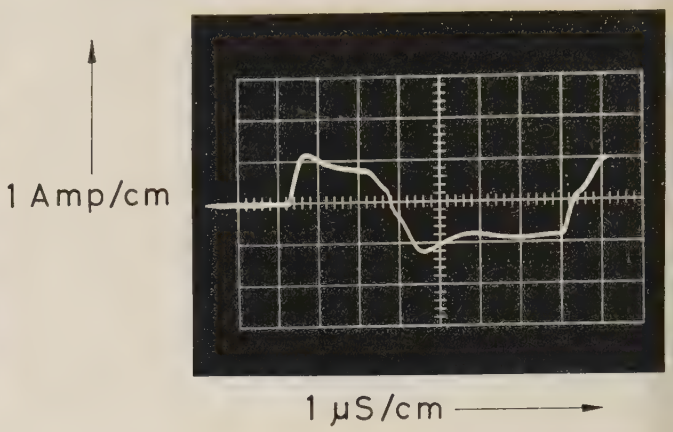


Fig. 13—Drive current for the second core in a two-core switch for a double-coincidence store.

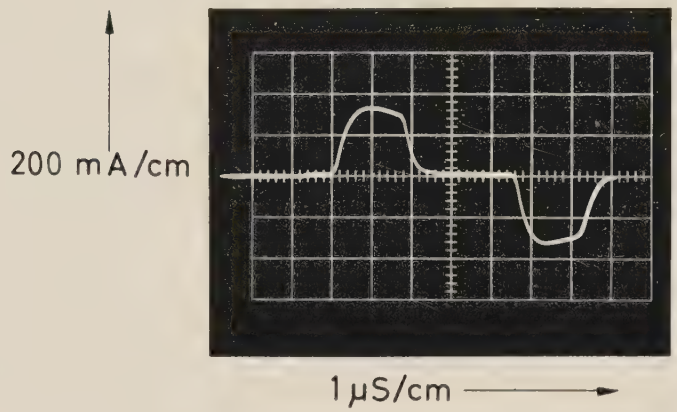


Fig. 14—Output from a transistor-driven two-core switch, driving a selection line in a double-coincidence store. The load contained 1000 cores (10 planes in series); the total impedance was 22 μh in series with 3.5 ohms.

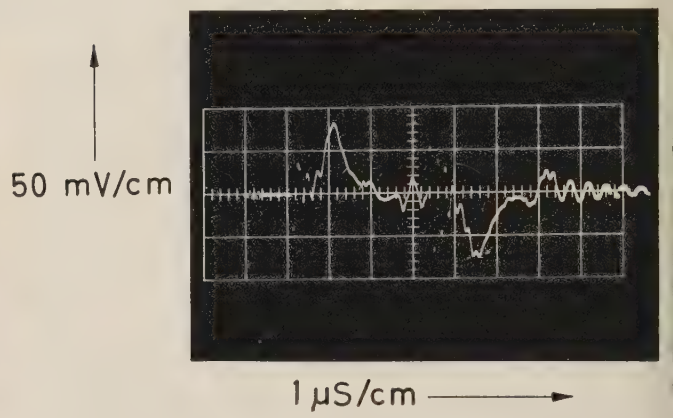


Fig. 15—Output from the sense winding of an 80×100 matrix driven by the two-core switch described in Section VII and Figs. 12–14. This output was taken via a screened 1:1 transformer.

Other arrangements of the two-core switch are possible. For example, Fig. 16 shows an X selection wire, driven by two cores, one in each of two switch matrices driven by coordinates u and v. Single-turn windings are shown for convenience, although in practice multiturn windings would be desirable. Core 1 at the selected position is driven by equal currents u_1 and v_1 in one matrix, and core 2 by equal currents u_2 and v_2 in the other. Each matrix has a dc bias winding (not shown) in addition to the coordinate lines; this is of magnitude $-u_1$ in the first matrix and $-u_2$ in the second.

The operation of this system is as follows. At $t=0$, u_1 and v_1 are applied, switching the core 1 and starting the output pulse. The dc bias in the second matrix prevents core 2 from switching. Nonselected cores in the first matrix remain at J (Fig. 8) or at A. The positive pulse is terminated by applying u_2 and v_2, nonselected cores in the second matrix remaining at J or A (Fig. 8). The negative output pulse begins when u_1 and v_2 ar switched off and ends when u_2 and v_2 are switched off. These waveforms are shown in Fig. 17 (next page).

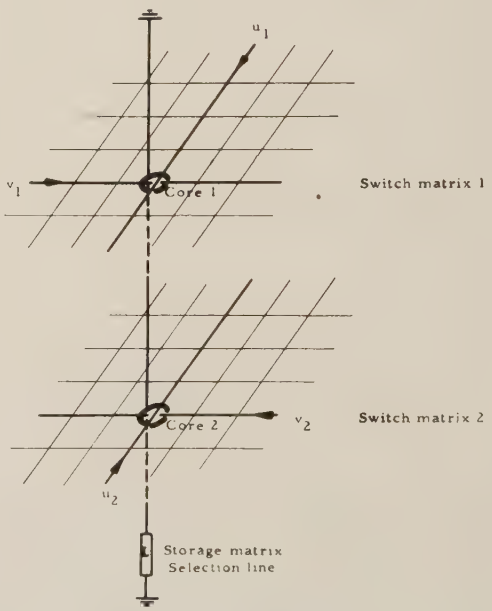


Fig. 16—Application of the two-core system to a double-coincidence matrix store.

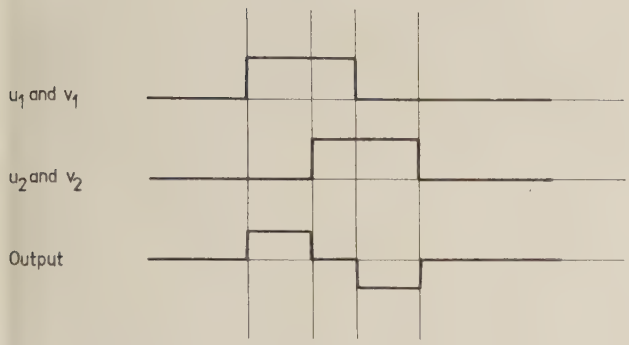


Fig. 17—Waveforms occurring in Fig. 16.

Various modifications of this simple scheme are possi-
e. Each matrix, considered alone, is very similar to
e conventional one-core scheme, and most of the vari-
ions that can be made to this apply to the two-core
heme. Several possibilities are to be found in the lit-
ature.[13]

Driving a Multiple-Coincidence Matrix Store

In this example, writing is performed by producing
e negative pulse of Fig. 1 only when a one is to be
itten. It follows that each plane must be driven
parately, since different planes will, in general, be re-
ired to store different information. Such an arrange-
ent is described briefly in Appendix I, and in more
tail in the literature.[16,19,20]

The circuit is shown in Fig. 18 and the waveform in
g. 19. The store, of which this circuit is a part, con-
ts of a number of planes using information in parallel
rm, one bit per plane. The circuit shown depicts a
ngle selection line of one plane. Windings *a* and *b* are
mmon to all core pairs in one coordinate of the se-
cted address; winding *d* is common to all core pairs
iving a single plane, and winding *c* is common to all
re pairs.

The positive output pulse is produced by the drive
rrents *a* and *b* as before, leaving both cores at *E* (Fig.
. If a one is to be written, a current pulse (shown
tted in Fig. 19) occurs in *d*. This does not affect the
res in the same plane at nonselected addresses, merely
iving them from *A* to *J* (Fig. 8), but it switches the
lected core to produce the required negative write
lse. If a zero is to be written, no current occurs in *d*
olid line in Fig. 19). The current in *c* insures that both
res are finally left at *A* (Fig. 8). It also serves to ter-
inate the write pulse, if that is present.

If the particular storage system permits, the *a* and *b*
rrents may be compounded from a double coinci-

[19] H. P. Schlaeppi and I. P. V. Carter, "Magnetic core memories
ng multiple coincidence," *Elektronische Rechenanlagen*, vol. 1, pp.
7–133; August, 1959. (In German.)
[20] H. P. Schlaeppi and I. P. V. Carter, "Submicrosecond core
mories using multiple coincidence," this issue, p. 192–198.

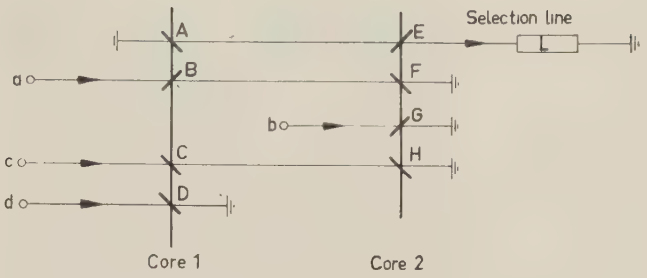


Fig. 18—Circuit for a two-core switch in a multiple-coincidence matrix.

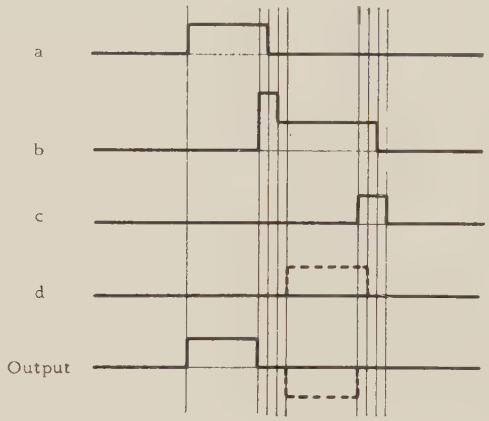


Fig. 19—Waveforms occurring in Fig. 18.

dence in a switch matrix, in a similar way to that described in the previous example (Section VII-A).

It should be noted that, for simplicity, all windings on core 1 (Fig. 18) have been assumed to have n_1 turns. In practice, of course, these can all be chosen to have suitable values and the corresponding windings on core 2 adjusted accordingly.

A model of and a driver for a 4:1 core store, with planes 101×101 has been described in detail elsewhere.[19,20] The selection-line waveforms for this model are displayed in Fig. 20 (next page), and details of its construction are given in Appendix VI. The cycle time is 1 μsec.

C. Other Uses of the Two-Core Switch

Apart from its use in driving storage matrices, the two-core switch possesses a combination of characteristics which may be useful in other applications. It is suitable for producing pulses of current of both signs in an inductive load. Its chief characteristics are its ability to act as a decoder (*e.g.*, in a switch matrix), its ability to accept badly shaped input pulses with very large amplitude variations, and its ability to produce any required sequence of pulses of various lengths and both signs, with great uniformity of amplitude.

A natural extension of the two-core switch is the multicore switch, in which the secondary windings of several cores are connected in series with an inductive load and used to begin, terminate, or vary the ampli-

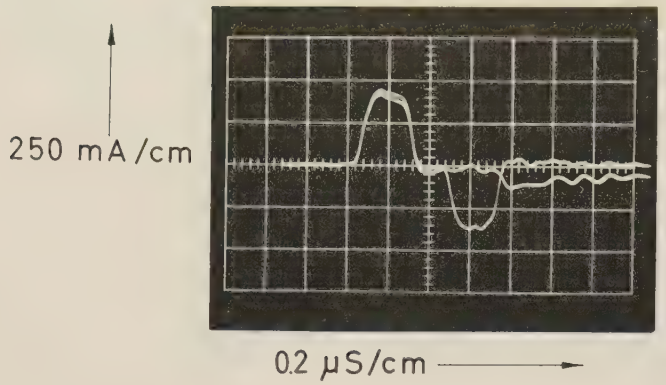


Fig. 20—Output from a two-core switch driving a 101×101 matrix at a selection ratio of 4:1. The sequence "read, write 1" is superposed on the sequence "read, write 0" (in which the second, negative pulse is missing). The slight difference in amplitude between the "read" pulses in the two cases is caused by the flux absorbed when the selected memory core is switched. Further details will be found in the literature.[19,20]

tude of the load current. The design of such a system is quite straightforward, following the principles established in Section IV and Appendix II of this paper. The change of load current caused by switching any core is given by (9), provided that the input current satisfies an equation of the form of (13) (modified to include any load current already flowing) and that the other cores are suitably biased to prevent them from being switched. A current established by switching a given core can be terminated by reversal of the same core (as suggested in the first part of Section IV), with some inaccuracy because of losses during the pulse; or, as in the two-core system, it can be terminated with another core, in which case the losses can be compensated by suitable choice of core windings. Although the current change caused by a given core is, of course, constant, different cores can be wound to cause different current changes.

VIII. Conclusions

A new way of using switch cores to drive magnetic core stores has been described which does not require added resistors in series with each selection line. The functions of decoding and transforming are performed as before. The load to be driven must be a fair approximation of a constant inductance (not necessarily linear).

At the cost of using twice the number of switch cores, the new switch offers a number of advantages. Pulse shaping and amplitude regulation are performed by the cores, thus permitting the use of crude and simple unregulated driving circuits. Very much less energy is required, and in particular the drivers dissipate much less power. The possibility of inhibiting the write pulse makes multiple coincidence storage very attractive where short cycle times (less than a microsecond) are required; the switch can be driven arbitrarily hard to achieve the necessary fast waveforms. One application of these principles has been the construction of address decoding and driving circuits for a double coincidence store of 10 planes, each 80×100, using only 34 transistors, two transformers and no diodes, with a cycle time of 9 μsec. A model of a 4:1 store with planes 101×101 has been built with a cycle time of 1 μsec. Quite apart from its usefulness as a storage-matrix driver, the switch offers a unique combination of useful properties for other applications.

Appendix I

Magnetic Core Storage

In the conventional magnetic core matrix, the core selected is at the intersection of two selection wires, each carrying a current, say I_d, such that $I_d < I_c < 21_d$, where I_c is the coercive current.

In a typical multiple-coincidence matrix (such as that described in the literature[16,19,20]), the selected core is linked by two additional selection wires, in such a way that none of the other cores of the matrix is threaded by more than one of these four selection lines. If each of these wires also carries a current I_d, then the selected core is linked by a total current $4I_d$, giving a "selection ratio" of 4:1.

If an additional "bias" wire threading all the cores in the matrix carries a current $-I_d$ and the four selection wires each carry a current $+21_d$, then all the non-selected cores are linked by a current of $\pm I_d$ and the selected core by a current $+71_d$. This gives a selection ratio of 7:1.

Appendix II

Detailed Calculation for the Two-Core Switch

At $t=0$, a current I_1 is applied to windings B and D (Fig. 7) of cores 1 and 2, and a current i_2 is induced in windings A and C. We have at time $t=t_1$:

$$\text{Flux change in core 1} = n_1\Phi + n_1^2kI_1 - n_1^2ki_2(t_1)$$

$$\text{Flux change in core 2} = n_2^2kI_1 - n_2^2ki_2(t_1)$$

$$\text{Flux change in load} = -Li_2(t_1)$$

[provided that $t_1 \ll L/R$ and (13) is satisfied].

Since the total flux change is zero, we have:

$$n_1\Phi + n_1^2kI_1 - n_1^2ki_2(t_1) + n_2^2kI_1 - n_2^2ki_2(t_1) - Li_2(t_1) = 0$$

$$\therefore\ i_2(t_1)(L + n_1^2k + n_2^2k) = n_1\Phi + kI_1(n_1^2 + n_2^2).$$

Writing $L^* = L + n_1^2k + n_2^2k$, we have

$$L^*i_2(t_1) = n_1\Phi + kI_1(n_1^2 + n_2^2). \tag{34}$$

At $t=T$, i_2 has decayed to a value $i_2(T)$, owing to a flux loss δ caused by dissipation in the load resistance and the presence of irreversible nonlinearitites in the load. Hence,

$$i_2(T) = i_2(t_1) - \frac{\Phi}{L^*}. \tag{35}$$

Also at $t=T$, a step function of current I_3 is applied such that (15) is satisfied. The current i_2 then changes to $i_2(T+t_1)$, and we have

Flux change in core 1 $= -n_1^2k(i_2(T+t_1) - i_2(T))$

Flux change in core 2 $= -n_2\Phi - n_2^2k(I_3 + i_2(T+t_1) - i_2(T))$

Flux change in load $= -L(i_2(T+t_1) - i_2(T))$.

Since the total flux change is zero, we have

$$n_1^2k(i_2(T+t_1) - i_2(T)) + n_2\Phi + n_2^2k(I_3 + i_2(T+t_1) - i_2(T)) + L(i_2(T+t_1) - i_2(T)) = 0.$$

Hence, from (34) and (35),

$$\begin{aligned}(T+t_1)\cdot L^* &= i_2(T)\cdot L^* - n_2\Phi - n_2^2kI_3 \\ &= i_2(t_1)L^* - \delta - n_2\Phi - n_2^2kI_3 \\ &= \Phi(n_1 - n_2) + kI_1(n_1^2 + n_2^2) \\ &\quad - kI_3\cdot n_2^2 - \delta. \end{aligned} \tag{36}$$

At $t=T+t_1$, current I_1 is switched off. i_2 then changes to $i_2(T+2t_1)$, and we have

Flux change in core 1 $= -n_1^2kI_1 - n_1^2k(i_2(T+2t_1) - i_2(T+t_1))$

Flux change in core 2 $= -n_2^2kI_1 - n_2^2k(i_2(T+2t_1) - i_2(T+t_1))$

Flux change in load $= -L(i_2(T+2t_1) - i_2(T+t_1))$.

Since the total flux change is zero, we have

$$n_1^2k + n_2^2k)(I_1 + i_2(T+2t_1) - i_2(T+t_1) + L(i_2(T+2t_1) - i_2(T+t_1)) = 0.$$

Hence, from (36),

$$\begin{aligned}i_2(T+2t_1)\cdot L^* &= i_2(T+t_1)L^* - kI_1(n_1^2 + n_2^2) \\ &= \Phi(n_1 - n_2) - kI_3\cdot n_2^2 - \delta. \end{aligned} \tag{37}$$

At $t=T+2t_1$, current I_3 is switched off. i_2 then changes to $i_2(T+3t_1)$ and we have

Flux change in core 1 $= -n_1^2k(i_2(T+3t_1) - i_2(T+2t_1))$

Flux change in core 2 $= n_2^2k(I_3 - i_2(T+3t_1) + i_2(T+2t_1))$

Flux change in load $= -L(i_2(T+3t_1) - i_2(T+2t_1))$.

Since the total flux change is zero, we have

$$n_2^2k(T_2'(T+3t_1) - i_2(T+2t_1)) + n_2^2k(-I_3 + i_2(T+3t_1) - i_2(T+2t_1)) + L(i_2(T+3t_1)) - i_2(T+2t_1)).$$

Hence, from (37),

$$\begin{aligned}i_2(T+3t_1)\cdot L^* &= i_2(T+2t_1)\cdot L^* + n_2^2kI_3 \\ &= \Phi(n_1 - n_2) - \delta. \end{aligned} \tag{38}$$

If this current is required to be zero, we have

$$\Phi(n_1 - n_2) = \delta \tag{39}$$

which gives the same choice of n_2 as that given by (14), bearing in mind that, in the approximation used in that case, $L=L^*$. It should also be noted that the exact value of the time constant shown in (12) as L/R is, in fact, L^*/R [*cf.*, (10), where only one core is present].

Appendix III

The Voltage-Driven Conventional Switch

A complete design procedure for this case has been described in the literature.[8,14] In contrast, the following simplified approach does not consider the problem of optimum design, but is concerned with the expressions for rise and decay times and pulse length.

We shall take

$$\begin{aligned}e(t) &= 0 \quad \text{for} \quad t < 0 \\ &= E \quad \text{for} \quad t > 0.\end{aligned}$$

As before, the core is initially at state A (Fig. 3) with no current flowing. When the voltage is applied, we now have [from (2)]

$$E = n\frac{d\phi}{dt} = i_2R + L\frac{di_2}{dt}. \tag{40}$$

Hence, we have a constant rate of change of flux, and for $i_2(t)$ we obtain the well-known solution

$$i_2(t) = \frac{E}{R}(1 - e^{-(R/L)t}). \tag{41}$$

The rise time of this current pulse (0–90 per cent) must be equal to τ; *i.e.*,

$$2.3\frac{L}{R} = \tau. \tag{42}$$

The peak current value I_2 is now

$$I_2 = \frac{E}{R}. \tag{43}$$

During the pulse, the primary current will in theory be

$$i_1 = i_2 + \frac{J_c}{n}, \tag{44}$$

but in practice, i_1 will be larger than this because of the finite switching speed of the core. Alternatively, we could say that (44) is always true, but that the effective value of J_c is larger than the quasi-static value when the core is being switched. Experiments show that $i_1(t)$ has a peak near the beginning of the pulse (depending on the rise time of e), then decreases, and increases again when the core is about half-switched. When the core reaches C, i_1 will increase rapidly, the core being driven into saturation in order to satisfy (40). In practice, the source will be capable of supplying currents only up to a certain limit, I_m; when this limit is reached, the source ceases to be a voltage source and becomes instead a current source (*e.g.*, if the source is a pentode or a transistor, its operating point has risen above the knee of the char-

acteristic). When this happens, the analysis of Section IV applies and i_2 decays exponentially, as before, with a time constant $(L+n^2k)R=\tau'$.

Since i_2 now has a finite rise time, we must decide how to define T. For simplicity, we shall here define it by (6) as before, *i.e.*,

$$T = \frac{n\Phi - I_2 L}{E} \tag{45}$$

which is easily seen to be shorter than the actual core-switching time [which is $n\Phi/E$ by (9)] by the amount $L/R=\tau/2.3$ [from (42)].

Appendix IV

The Voltage-Driven Two-Core Switch

Given L, I_2, T and τ, we choose

$$E_B = \frac{LI_2}{\tau} \tag{46}$$

$$R_B = 0.75\frac{E_B}{I_2} \tag{47}$$

and

$$C_B = \frac{\tau}{R_B}\cdot \tag{48}$$

We neglect stray inductance and capacity, and then have the equivalent circuit of Fig. 21. We also neglect the coercive current of the core, and then, before the core saturates, we have

$$R_B C_B \frac{di_R}{dt} = i_L - i_R \tag{49}$$

$$E_B = L\frac{di_L}{dt} + R_B i_R \tag{50}$$

where $i_R(t)$ is the current in R_B and $i_L(t)$ is the current in L.

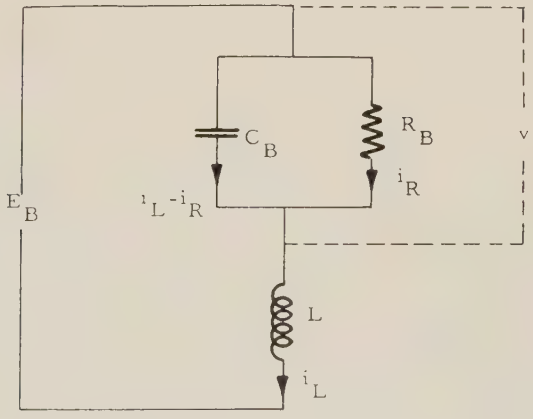


Fig. 21—Equivalent circuit of Case *B*.

Hence,

$$E_B = L\frac{di_R}{dt} + LR_BC_B\frac{d^2i_R}{dt^2} + R_Bi_R$$

and

$$\frac{d^2i_R}{dt^2} + \frac{1}{\tau}\frac{di_R}{dt} + \frac{3}{4\tau^2}i_R = \frac{I_2}{\tau^2} \tag{51}$$

since from (46), (47) and (48),

$$LC_B = \frac{E_B\tau}{I_2}\cdot\frac{\tau}{R_B} = \frac{4\tau^2}{3}\cdot \tag{52}$$

The solution of (51) is readily found to be

$$i_R = \frac{4}{3}I_2\left\{1 - e^{-t/2\tau}\left(\frac{1}{\sqrt{2}}\sin\frac{t}{\tau\sqrt{2}} + \cos\frac{t}{\tau\sqrt{2}}\right)\right\} \tag{53}$$

when the initial conditions $i_R=di_R/dt=0$ at $t=0$ are inserted. Hence, using (49), we also have

$$i_L = \frac{4}{3}I_2\left\{1 + e^{-t/2\tau}\left(\frac{1}{2\sqrt{2}}\sin\frac{t}{\tau\sqrt{2}} - \cos\frac{t}{\tau\sqrt{2}}\right)\right\}. \tag{54}$$

The voltage v across R_B is then given by

$$v = i_RR_B = i_R\cdot\frac{3E_B}{4I_2}$$

$$= E_B\left\{1 - e^{-t/2\tau}\left(\frac{1}{\sqrt{2}}\sin\frac{t}{\tau\sqrt{2}} + \cos\frac{t}{\tau\sqrt{2}}\right)\right\}. \tag{55}$$

Graphs of i_L/I_2 and v/E_B as functions of t/τ are shown in Fig. 22. It will be noted that when $t=\tau$, $i_L=0.9I_2$ (thus confirming the design condition that the rise time 0–90 per cent should be τ). The core saturates when $t=1.14\tau$; at this time, $v=0.32E_B$.

After the core saturates, L is effectively zero and the transistor acts as a current source. The actual value of current I_m depends on the circuits driving the transistor; we shall suppose that $I_m=2I_L$ and later discuss the effects of other values of I_m.

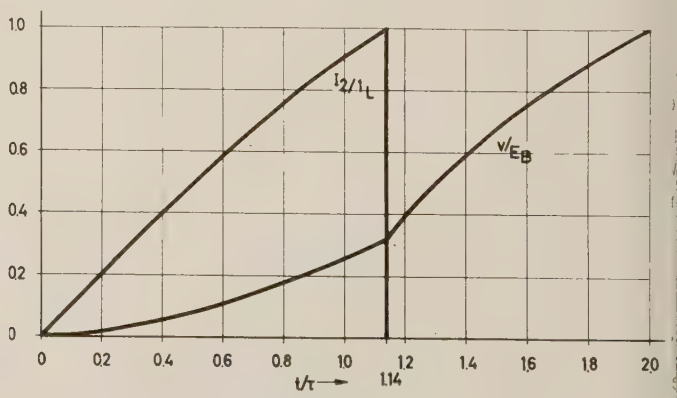


Fig. 22—Graph of I_2/I_L and v/E_B as function of t/τ.

We now have $L=0$ and $i_L=I_m=2I_2=$constant. ence, from (49) we have

$$\frac{di_R}{dt}+\frac{i_R}{\tau}=\frac{2I_2}{\tau}. \tag{56}$$

The solution of (55) is readily found to be

$$i_R = I_2(2 - 4.93\cdot e^{-t/\tau}) \tag{57}$$

nce at $t=1.14\tau$ we have

$$i_R = v/R_s = 0.319E_B/R_B = 0.425I_2.$$

ence, directly,

$$v = i_R R_B = E_B(1.5 - 3.70e^{-t/\tau}). \tag{58}$$

his is also plotted in Fig. 22. The transistor saturates hen $v=E_B$, *i.e.*, when $t=2.00\tau$.

The instantaneous power dissipated in the transistor uring this time is

$$I_m(E_B - v) = 2I_2E_B(3.70e^{-t/\tau} - 0.5) \tag{59}$$

d hence the total energy dissipated during this phase

$$2I_2E_B\int_{1.14\tau}^{2.00\tau}(3.70e^{-t/\tau} - 0.5)dt = 0.50I_2E_B\tau. \tag{60}$$

we choose I_m other than $2I_2$, we obtain different alues for the constant term of (60), varying between 91$I_2E_B\tau$ when $I_m=(4/3)I_2$ (the minimum necessary to sure that the transistor saturates) and $0.31I_2E_B\tau$ hen I_m is very large.

Appendix V

Energy Dissipation in a Core

Consider a core of magnetic path length l, and irre- rsible flux Φ webers per turn. Suppose this core has a inding of n turns.

The switching time t_s for the core to undergo a 90 r cent flux reversal under a constant current I_1 in the turn winding is found experimentally to be given by

$$t_s = \frac{lS}{nI_1 - J_0l}$$

here S (the switching coefficient) and J_0 are constants the core material and J_0l is app oximately equal to J_c.

When a constant voltage E is applied across the inding, the switching time (for 90 per cent reversal) is

$$t_s = 0.9\frac{n\Phi}{E}.$$

Adopting the usual rather crude approximation that ne current flowing is constant throughout this time, we ave

$$0.9\frac{n\Phi}{E} = \frac{lS}{nI_1 - J_0l}$$

$$\therefore\ nI_1 - J_0l = \frac{lSE}{0.9n\Phi}$$

and

$$I_1 = \frac{J_0l}{n} + \frac{lSE}{0.9n}.$$

$$\text{Total energy dissipation} = \frac{EI_1t_s}{0.9} = I_1n\Phi$$

$$= J_0l\Phi + \frac{lSE}{0.9n}. \tag{61}$$

Provided that E is constant, this result is independent of secondary currents flowing in other windings on the core.

Appendix VI

Details of the Circuits Used for Figs. 12–15 and 20

1) Figs. 12–15. In the notation of Fig. 11, $E_B=35$ volts, $C_B=0.22$ μf, $R_B=16.5$ ohms (first core) and 28 ohms (second core). The transistors were Telefunken type OD 6001. The cores used each consisted of two 3.8-mm-diameter Ferroxcube 6E1 cores wound together, giving $\Phi=1.36\times10^{-6}$ webers and $J_c=0.6$ ampere-turns, approximately. The primary windings each had two turns; the secondary windings had six turns on the first core and four turns on the second core. The primary windings of all 180 first cores were connected in series, and driven by two transistors via a 1:1:1 push-pull transformer; the primaries of the 180 second cores were similarly driven. As already described, the 180 core pairs were arranged in two matrices, and all except one core pair in each matrix were prevented from switching by bias currents supplied from 30 similar transistors.

2) Fig. 20. The circuit diagram is shown in Fig. 18. The storage matrix was 101×101, the storage cores being 2-mm-diameter Ferroxcube 6D1. Referring to Fig. 18, for the horizontal and vertical coordinates, each core was 2-mm 6B1, $n_1=4$ and $n_2=3$; for the slanting coordinates, each core consisted of 2×2-mm 6B1 wound together, $n_1=5$ and $n_2=4$. Vacuum-tube drivers were used. A single 2-mm 6B1 core has $\Phi=0.10\times10^{-6}$ webers and $J_c=0.4$ ampere-turns. Further details will be found in the literature.[19,20]

Acknowledgment

The work presented in this paper was performed at the IBM Research Laboratory, Zurich. The author wishes to thank H. P. Schlaeppi for his help and advice; thanks are also due the IBM Corporation for permission to publish this paper.

Submicrosecond Core Memories Using Multiple Coincidence*

H. P. SCHLAEPPI†, MEMBER, IRE, AND I. P. V. CARTER†

Summary—Memories using toroidal ferrite cores with cycle times less than a microsecond are described; the selection ratio is increased by the use of biasing and the multiple coincidence principles of Minnick and Ashenhurst.[1] It is shown that this mode of operation leads to important changes in the structure of the store; in particular, the classical core switch does not fulfill the new requirements. The "two-core switch" is then briefly described; it permits an elegant and economic solution of the problems arising at high selection ratios. Details of the design and operation of memories embodying these ideas are given; it is shown, for example, that standard core memory matrices can be used very efficiently at a selection ratio of 3:1 to achieve a cycle time of 2 microseconds. Further illustrations are given from a model of a 100×100 store operated at 4:1 and 7:1 selection ratios, and it is shown that a store of 10,000 8-bit characters with a cycle time of 0.25 microsecond is feasible.

Introduction

IN the classical magnetic core store, a core is selected by applying unit current pulses to two independent selection lines (Fig. 1). The selected core will thus receive two units of current, whereas an unselected core will receive at most only one unit of current. This type of memory is, therefore, said to operate at a selection ratio of 2:1. The switching time of the core is found to be given by

$$\tau_s = \frac{s}{2I_s - I_c}$$

s = core switching constant

I_c = threshold current

I_s = selection line current

$$I_s < I_c < 2I_s.$$

In the present paper, we consider the possibility of increasing the selection ratio, which yields either relaxed tolerances on cores and/or driving currents or increased memory speed due to the higher switching field (Fig. 2). Two methods of increasing the selection ratio are discussed: adding a bias winding to the memory plane, and adding sets of redundant selection wires to a plane.

In the classical 2:1 magnetic core memory, writing is performed by means of an inhibit current (Fig. 1). It is clear that if the selection ratio is larger than 2:1, this method of writing can no longer be used. Hence, the only remaining possibility of writing lies in modifying the current pulses on the selection wires themselves. Furthermore, since it is necessary to be able to write different information into each plane separately, the practice (standard in the classical 2:1 store) of connecting corresponding selection wires of several planes in series can no longer be used, and each plane must be separately driven. The use of magnetic core switches is indicated, but conventional switches working into re-

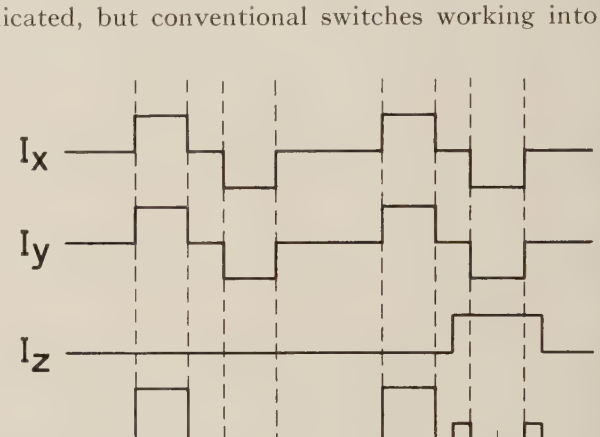


Fig. 1—Operation of conventional 2:1 memory.
I_x, I_y = Selection line currents
I_z = Inhibit current applied to all cores in the plane
ΣI = Net current waveform received by the selected core.

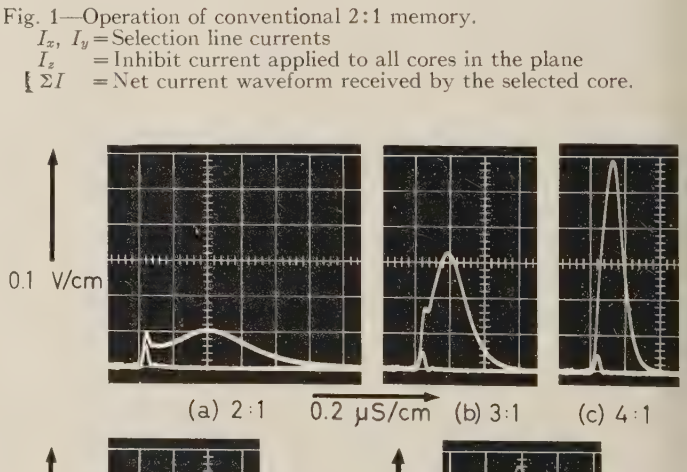


Fig. 2—Memory core switching at various selection ratios. (a)–(d) Undisturbed ones and zeros from an 80/50 core. (e) The same, from a 50/30 core. Note the signal-to-noise ratio and the signal amplitude.

* Received by the PGEC, February 9, 1960. This paper was presented at the 1960 Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 10–12.

† Research Laboratory Zurich, IBM Corp., Adliswil-Zurich, Switzerland.

[1] R. C. Minnick and R. L. Ashenhurst, "Multiple coincidence magnetic storage systems," *J. Appl. Phys.*, vol. 26, pp. 575–579; May, 1955.

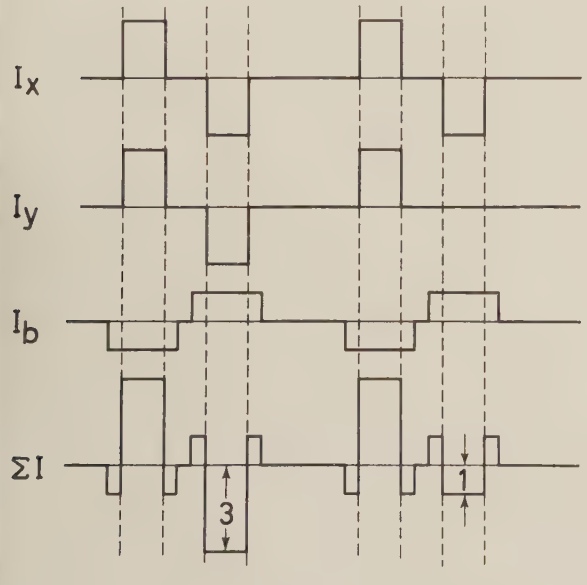
stive loads cannot be used. Therefore, a new technique using magnetic cores as switches has been developed, he "two-core switch"[2]) and its application in this case described.

Minnick and Ashenhurst[1] have solved the theoretical roblem of how to achieve large selection ratios by lding sets of redundant coordinates. We have inves-gated the switching and driving problems peculiar to ultiple coincidence stores, and have verified experi-entally the feasibility of such stores.

A model of a 100×100 core 7:1 memory using two-ore switches was constructed with standard 80/50-mil ores for both storage and switching. Constructional etails and results obtained with this model are dis-ussed; cycle times of less than a microsecond are early indicated, which could be further reduced if 0/30-mil cores were used.

Methods of Increasing the Selection Ratio

A standard core memory plane having two orthogonal ts of selection lines may be operated with a selection tio of 3:1, by supplying each selection line of a se-cted core with two units of current instead of one, nd by opposing both the read and write selection ulses by bias pulses of one unit magnitude (Fig. 3). ince the inhibit winding is no longer useful for writing, can serve for carrying the bias pulses. The selected ore receives three units of current, while the other cores the plane receive either plus one unit or minus one nit of current. Thus, the selection ratio has been in-reased from 2:1 to 3:1. The cycle time for such a ore can be reduced to approximately half that of the conventional 2:1 store (Fig. 2). Writing must be performed by suppressing the write pulse in at least one of the two sets of selection lines (Fig. 3). (It is not necessary to suppress the writing pulse in both selection lines.) A block diagram showing a possible arrangement of such a store is shown in Fig. 4. The X selection lines are connected in series throughout the stack of planes, as usual in the 2:1 case; the Y selection lines are separately driven for each plane in order to permit writing. The operation of the core switches used will be discussed in the next section.



g. 3—Operation of a double coincidence memory at 3:1 selection ratio.
I_x, I_y = Selection line currents
I_b = Bias current, applied to all cores in the plane
ΣI = Net current waveform received by the selected core.

[2] I. P. V. Carter, "A new core switch for magnetic matrix stores d other purposes," this issue, p. 176.

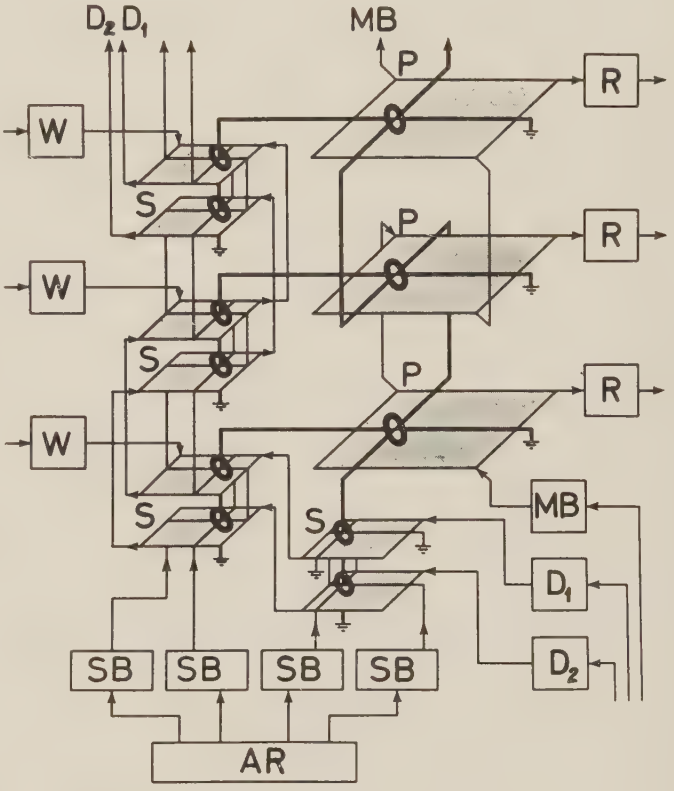


Fig. 4—Double coincidence memory with 3:1 selection ratio.
P = Memory plane
S = Two-core switches
D_1, D_2 = Switch drivers
AR = Address register
SB = Switch bias drivers
MB = Memory bias drivers
W = Write inhibit drivers
R = Read amplifiers.

An alternative method of increasing the selection ratio is to add extra selection lines. Such an arrangement is shown in Fig. 5, which shows the standard core plane to which two extra sets of selection wires have been added. The arrangement must be such that no two wires threading any one core have any other core in common; it has been shown[1] that this condition puts only slight restrictions on the matrix format. These extra sets of selection lines are redundant, in the sense that the core is already selected by the choice of the first two (X and Y) selection lines. The choice of the extra two selection lines (U and V, or "slanting") which must

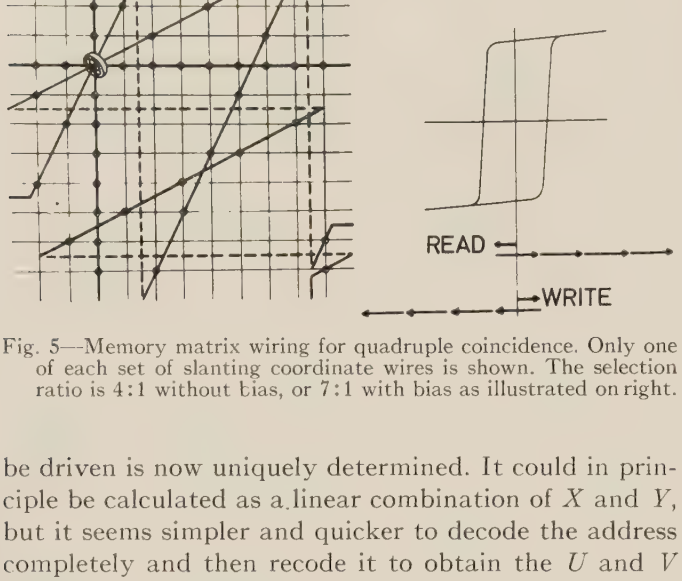


Fig. 5—Memory matrix wiring for quadruple coincidence. Only one of each set of slanting coordinate wires is shown. The selection ratio is 4:1 without bias, or 7:1 with bias as illustrated on right.

be driven is now uniquely determined. It could in principle be calculated as a linear combination of X and Y, but it seems simpler and quicker to decode the address completely and then recode it to obtain the U and V coordinates. This can be done easily by using a magnetic core matrix which is physically identical to a memory plane. The X and Y lines from this plane are driven in the usual way, as selected by the address register; the selected core is switched, and it induces pulses in the two correct slanting coordinate lines. These pulses trigger the driving circuits for the slanting coordinates directly. This translator matrix can be seen in the lower part of Fig. 6, which shows the block diagram of such a store.

With a total of four sets of selection lines, as shown in Fig. 5, a selection ratio of 4:1 is possible. It can be increased to 7:1 by adding a bias current and doubling the currents in the selection lines, exactly as before.

The Two-Core Switch

If core switches are used in the conventional way with added load resistors, it is not possible to obtain the fast pulses needed without prohibitive expenditure of power. Also, such switches do not lend themselves readily to the necessary suppression of the writing pulse when it is required to write a zero. A new technique of using magnetic cores as switches has, therefore, been developed and is fully described elsewhere.[2] In this switch, the load resistance is reduced to a minimum. The switch then works on the principle of flux switching, in that the switch core is completely switched at the beginning of the pulse and the secondary current is then exclusively determined by the flux linkage of the switch core and the load inductance. The pulse is terminated by a second switch core operating in opposition to the first one. A write pulse can be made by resetting the two cores, one after the other, as before; or, it can be suppressed by resetting the two cores simultaneously. Since this switch provides inherent regulation, the switch cores can be driven arbitrarily hard in order to achieve the necessary fast rise times. Since the load resistance is a minimum, the energy loss is also a minimum and the power requirements are much smaller than with the conventional switch. The shape of the hysteresis loop flanks no longer influences the current waveform, and the flux linkage is also greatly reduced; thus, ferrite cores may be used.

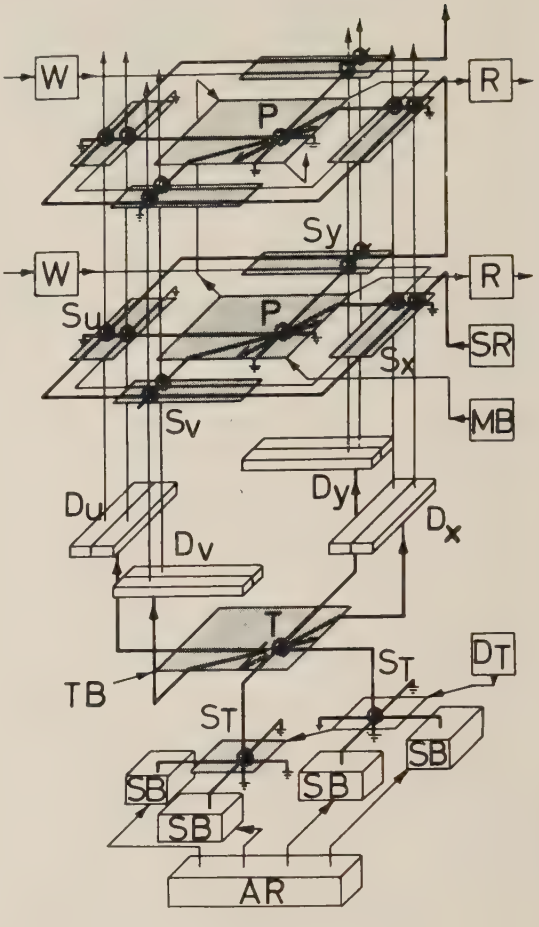
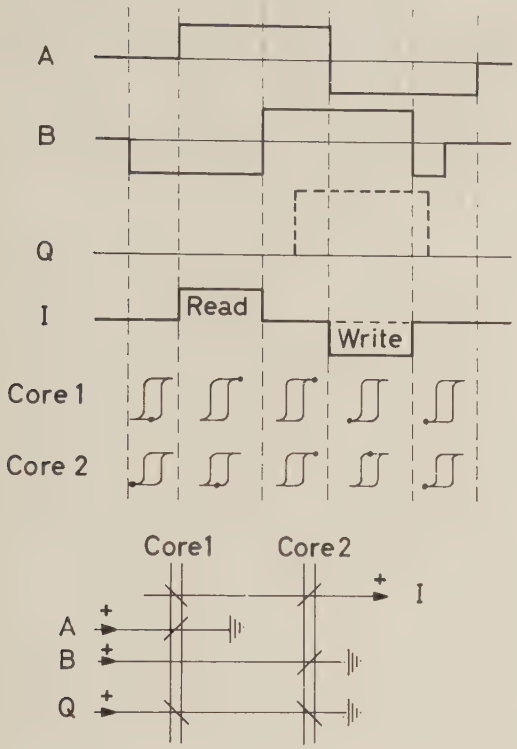


Fig. 6—Block diagram of 7:1 store.

AR	= Address register	
S_T	= Switches	for translator matrix
SB	= Switch bias drivers	for translator matrix
D_T	= Switch drivers	for translator matrix
T	= Translator matrix	
TB	= Translator matrix bias	
$D_uD_vD_zD_y$	= Switch drivers	for memory matrix
$S_uS_vS_xS_y$	= Two-core switches	for memory matrix
SR	= Switch reset driver	for memory matrix
W	= Write drivers	for memory matrix
P	= Memory matrix	
MB	= Matrix bias driver	
R	= Read amplifier.	

The application of this switch to the 3:1 store shown in Fig. 4 is illustrated in Fig. 7. Address information is delivered to the X and Y switch matrices by means of bias currents impeding the switching of all but the selected switch cores—a procedure used also with conventional switch matrices.[3] The drive lines of all the two-core switches are series connected and are supplied from a common pair of drivers (A and B in Fig. 5). Writing a "zero" into a particular plane is achieved by the "write inhibit" driver W of that plane, which simultaneously resets both cores of the switch, previously

[3] J. A. Rajchman, "A myriabit magnetic-core matrix memory," Proc. IRE, vol. 41, pp. 1407–1421; October, 1953.



ig. 7—Two-core switch circuit and waveform diagram (3:1 memory).
A and *B* = Drive currents
Q = Write inhibit current (suppresses "write 1" pulse in order to write 0)
I = Output current in selection line.

elected by the read drive, before the arrival of the second part of the drive waveforms which would otherwise eset the two cores one after the other to make a "write " pulse (Fig. 7).

For the 7:1 case, the circuit is shown in Fig. 8. The witch drives a single selection line of one plane in the tore. Windings *a* and *b* are common to all switches in ne coordinate of the selected address; winding *d* is ommon to all switches driving the selection lines of a ngle plane, and winding *c* is common to all switches in ne memory stack. The positive read pulse is produced y the drive currents *a* and *b* as before, leaving both ores in the state of positive remanence. The waveorms are shown in Fig. 8. The purpose of winding *F* is o prevent the second core being prematurely switched y the output current in *E*. If a one is to be written, a urrent pulse (shown dotted in Fig. 8) occurs in *d*; this oes not affect the cores in the same plane at nonselected ddresses (merely driving them further into saturaion), but it switches the selected core to produce the equired negative write pulse. If a zero is to be written, o current occurs in *D* (solid line in Fig. 8). The current n *C* insures that both cores are finally left at negative emanence. It either terminates the write pulse if it ere present, or it resets both cores simultaneously. It ill be noted that all the driving currents are unidirecional; this helps to keep the drivers, the number of hich is relatively large in such a memory, as simple s possible.

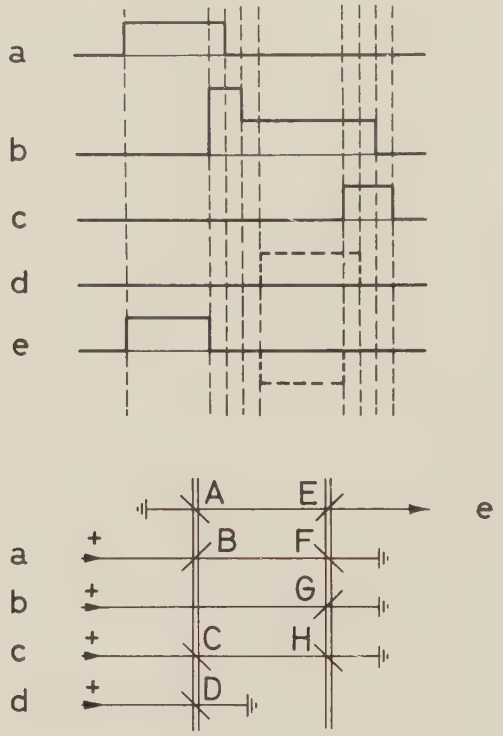


Fig. 8—Two-core switch circuit and waveform diagram (7:1 memory).
a and *b* = Drive (forms Read pulse)
c = Reset (terminates Write pulse, or resets both cores)
d = Write (forms rising edge of Write pulse)
Windings *A*, *B*, *C* and *D* have n_1 turns
Windings *E*, *F* and *G* have n_2 turns
Winding *H* has $2n_2$ turns.

The practical design of the switch is very simple. From the known inductance of the selection line and the known required selection line current, the flux linkage of the switch core can be calculated.

$$n\Phi = LI_s$$

n = number of secondary turns

Φ = available flux from core

L = load inductance.

The requirements on the switch core material are that it should have as small a permeability as possible in the saturation region and a low coercive force. Its squareness ratio (B_r/B_s) must be as high as possible, but the shape of the hysteresis loop flanks is of no interest. The switching constant of the material should be low; this, together with the choice of a low coercive force, reduces the losses in the core and enables fast switching times to be achieved without an unnecessarily large number of turns on the core. In practice, it has been found that standard 80/50 memory cores behave well as switch cores in this mode of operation.

Since it is not necessary to standardize the driving currents, the design of transistor drivers for these switch cores presents few problems. A suitable circuit is shown in Fig. 9. During the rise time of the output pulse. the transistor is saturated and the whole supply voltage

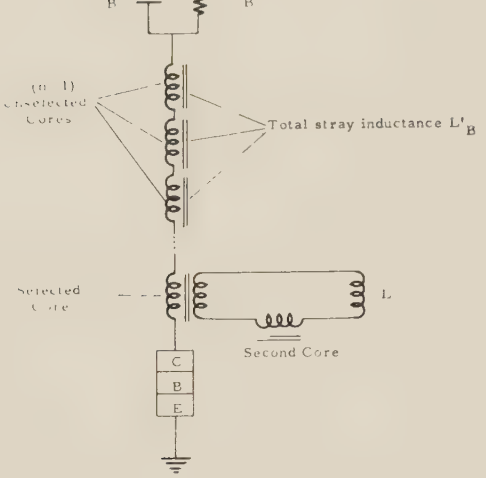


Fig. 9—Transistor driver for two-core switch. For simplicity, the turns ratio on the switch cores is assumed to be 1:1.

is applied across the selected switch core. The rise time of the output current pulse is then

$$\tau_r = \frac{LI_s}{E_B}$$

E_B = supply voltage

τ_r = rise time (0–90 per cent).

If the turns ratio on the core is other than 1:1, the equation must be correspondingly modified. When the switch core saturates, thus presenting a short circuit to the transistor, the condenser C_B charges rapidly, preventing excessive power dissipation in the transistor. The resistor R_B limits the current in the transistor during the pulse and the time constant R_BC_B is made equal to τ_r. In a complete 7:1 store, only 8 R_BC_B pairs will be required. It can be shown that the energy dissipation per cycle in the transistor with such a circuit is equal to $\frac{1}{4}\ \tau_r I_m E_B$, where I_m is the maximum current obtainable from the transistor before it saturates. A full description of this circuit, including the effect of stray inductance, has been given elsewhere.[2]

The translator matrix can also be driven by means of core switches working in the mode described for the two-core switch; since only a unipolar drive pulse is required for the translator, the translator switches need have only a single core.

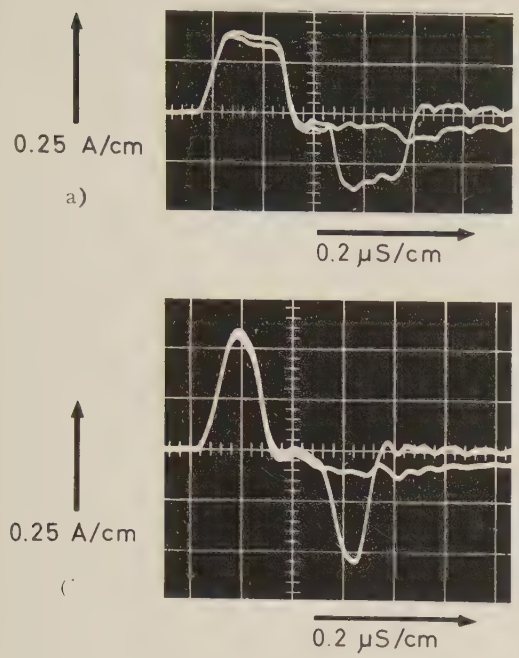
An Experimental 100×100 Memory Model

In order to test the ideas presented in this paper, a model of a memory plane 100×100 was built using standard 80/50-mil cores made from Philips 6D1 material. In order to satisfy the necessary conditions of matrix format,[1] the actual size of the matrix was 101 ×101. The measured inductance of the selection lines on this matrix was 0.75 microhenry for the orthogonal X and Y coordinates, and 1.85 microhenries for the slanting U and V coordinates. For the 4:1 case, therefore, where a selection line current of 0.4 ampere was required, a flux change of 3×10^{-7} volt seconds was required for the X and Y lines and a flux of 7.5×10^{-7} volt seconds for the U and V lines. As switch-core, a 80/50-mil core of Philips 6B1 material was chosen, which had a switchable flux of approximately 10×10^{-8} volt seconds. For the X and Y lines, four turns round one such core was taken for the secondary winding, and for the U and V lines, five turns around two such cores wound together was used. The output from this switch is shown in Fig. 10(a). A cycle time of 1 microsecond is clearly easily feasible. The slight change in amplitude of the read pulse visible in the figure is due to the flux absorbed by a memory core when switching.

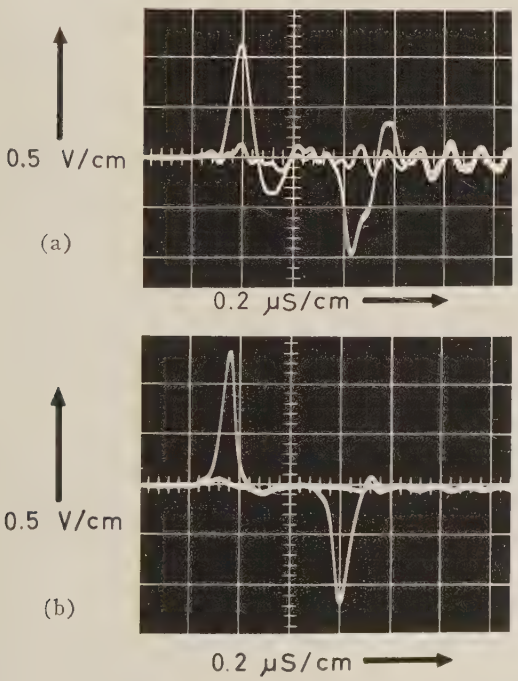
With the construction of this switch, it became immediately apparent that an important factor in the design of such stores (with cycle times of a microsecond or less) is the heating effect in the switch cores and the memory cores themselves. With forced air cooling it is possible to switch a standard 80/50-mil core up to about 1 or 2 megacycles without excessive overheating, but above that frequency some other form of cooling is required. Experiments have shown that standard 80/50-mil cores can be switched at over 2 megacycles without undue heating if immersed in pentane whose boiling point is 35°C. Local boiling of the liquid prevents overheating of the core. This method of cooling could readily be applied to a complete memory, by immersing both memory cores and switch cores in a suitable coolant, in a sealed container. It would not be necessary to provide special cooling for the liquid itself, since the total power dissipation is extremely small.

For the 7:1 case, the selection line current can be as large as 0.8 ampere. However, the switches on our model were designed for 0.6 ampere only, since such a value would permit cycle times of $\frac{3}{4}$ microsecond, *i.e.*, a repetition rate of 1.3 megacycles which is near the limit of operation with air cooling. The waveform from these switches is shown in Fig. 10(b). The switches were the same as before, except that cores with a switchable flux of approximately 14×10^{-8} volt seconds were used instead of the 6B1 cores. The waveforms shown indicate clearly the possibility of a $\frac{3}{4}$-microsecond cycle time. It will be noted that the pulses are far from being rectangular; this is no disadvantage in a multiple coincidence store, since the tolerances are much more relaxed than in the classical 2:1 store, and it saves a great deal of driving power, as will be shown later.

The output from the sense wire of the matrix is shown in Fig. 11(a). Both the signal-to-noise ratio and the actual amplitude of the signal are good. The output shown was taken via a 1:1 screened transformer, and the signal was thereby slightly differentiated. By increasing the turns ratio of this transformer, considerable further amplification of the signal is possible. The actual signal



ig. 10—Selection currents for quadruple coincidence memory. The program was: Read, Write "1," Read, Write "0." The "read" pulse amplitude changes slightly when a memory core switches. (a) 4:1. (b) 7:1.



ig. 11—Waveforms from 7:1 memory. The program was: Read, Write "1," Read, Write "0." (a) Read signal from sense line, via 1:1 screened transformer. (b) Signal from selected core alone.

om the selected core itself (without the sense wire) is nown in Fig. 11(b) for comparison.

Analysis of Dependence of Drive Power on Cycle Time

It is interesting to study the variation of driver rating (*i.e.*, peak voltage×peak current—this figure is, of course, much larger than the actual power dissipated in either switches or drivers) with cycle time. We have

$$\text{Peak voltage} = \frac{LdI_s}{dt} = \frac{LI_s}{\tau_r} \qquad \text{(approximately)}.$$

Hence the driver rating is

$$P = \frac{LI_s^2}{\tau_r}.$$

Now

$$\tau_s = \frac{s}{\frac{7}{2} I_s - I_c} = \frac{2s}{7I_s} \qquad \text{(approximately)}.$$

Hence

$$P = \frac{KL}{\tau_r \tau_s^2}$$

where

$$K = \frac{49}{4s} \qquad \text{is a constant.}$$

Also the cycle time is

$$T = 2\tau_s + 3\tau_r + 2\tau_0$$

where τ_0 is the gap between pulses. This assumes that the core switches throughout the time that the drive pulses are above 75 per cent peak amplitude.

If τ_0 is constant, it can be shown that for constant T, P is a minimum when

$$\tau_s = 3\tau_r.$$

In this case we have

$$T = 3\tau_s + 2\tau_0,$$

and the cycle time will in practice be approximately four or five times the memory core switching time. This criterion suggests that both the 4:1 and the 7:1 models described are not very close to optimum working, since in both cases T/τ_s is near ten.

If τ_0 is dependent on τ_r (as would be the case with nearly perfect driving circuits), then the optimum value of τ_s/τ_r increases; for example, if $\tau_0 = 2\tau_r$ then $\tau_s/\tau_r = 7$ for minimum P. This case is of little practical interest, since it means that very small values of τ_r are needed, leading to practical difficulties which outweigh the very small economies in driver rating.

Whatever the dependence of τ_0 on τ and/or T, the minimum value of P will vary inversely as T^3. This effect is serious, and various methods of reducing driver rating must be considered.

Possible Improvements

Both individual and collective driving power can be reduced by increasing the selection ratio, but it does not seem very practicable to go beyond 7:1.

A reduction in memory core size would reduce the selection line currents required, and also allow a smaller matrix and hence reduced selection line inductance. The smallest size of readily available ferrite core is the 50/30-mil core; Fig. 2 shows that switching of such a core at 7:1 selection ratio is accomplished in 0.05 microsecond with 0.6-ampere selection current.

A reduction in selection line inductance can be made by "folding" the matrix. From Fig. 5 it is clear that if l is the length of the matrix edge, then the length of an X wire is $2l$ (including return lead) and that of a U wire is $(3+\sqrt{5})l$. If the matrix is folded in half about the y axis, the ends of the X wire come together and the total length is only l. Similarly, the length of the U wire is reduced to $(1+\sqrt{5})l$ and that of the V wire to $(2+\sqrt{5})l$.

A second folding operation about the X axis will reduce the wire length to l for the orthogonal and to $l\sqrt{5}$ for the slanting coordinates. In practice, these folding operations are achieved by wiring the four quarters of the matrix separately and by subsequent stacking. This would reduce the selection line inductance by about 40 to 50 per cent. Unfortunately, the presence of the slanting lines does not permit packing the cores as closely as is possible in 2:1 memories, so that the inductance component originating in the wire lengths themselves turns out somewhat larger than it would be if only two sets of selection lines were used.

A further reduction of inductance is achieved by sandwiching the selection wiring between grounded shields, and by using a wire gauge as heavy as possible.

For a 10,000 bit matrix of this compressed layout using ordinary 50/30/20 mil cores, a selection loop inductance of 0.55 μh is calculated. With a selection ratio of 7:1, a read-write cycle-time of 0.25 microsecond could be obtained using a read and write pulse width of 0.065 microsecond and rise and fall times of 0.02 microsecond. Given a selection current of 0.5 ampere, the driver power rating would be 7.2 volt-amperes per plane. The same core plane in the 4:1 mode would require a 0.9-volt-ampere drive rating at a cycle time of about half a microsecond.

A 10,000 word store operating with 4:1 or 7:1 selection requires 808 drivers. If driver transistors capable of dissipating 2.3 watts (maximum) are used, up to 8 planes can be driven at 4 megacycles using 7:1 selection ratio. Using 4:1 selection and working at 2 megacycles reduces the dissipation to 0.3 watts. Both figures are peak values, assuming the same drivers are selected every cycle.[4] Recent advances in the field of transistor technology seem to indicate that the driver transistor specifications can be met.

Improvements in the memory cores themselves could certainly effect substantial savings. The hyestersis loop criteria which dominate core design for 2:1 memories are no longer very stringent because of the high selection ratio, and, hence, greater emphasis could be laid on reducing the switching coefficient.

This discussion has assumed that the cycle time depends on the characteristics of the switch drivers, switches and storage matrix. It is clear from Fig. 6 that there will be a certain delay introduced by the translator matrix, which could well amount to a substantial fraction of the cycle time. However, once the switch drivers have been triggered from the translator, the address register can at once be altered to the next value (at a time roughly half way through the memory cycle), and thus the translator matrix delay can be accommodated in the second half of the preceding cycle.

Conclusion

The organization and operation of a 2-microsecond core memory using standard 80/50-mil core planes at a selection ratio of 3:1 and that of a $\frac{1}{2}$-microsecond memory using 80/50 cores in a multiple coincidence wiring giving a 7:1 ratio are described. Both systems depend on a new core switch, which is briefly discussed, and which permits write-pulse suppression and a large reduction in driving power. Further improvements indicate the feasibility of a $\frac{1}{4}$-microsecond memory using 50/30 ferrite cores and transistor drivers, with a capacity of 10,000 8-bit characters.

Acknowledgment

The authors wish to thank their colleagues at the IBM Research Laboratory Zurich for several helpful discussions, and the IBM Corporation for permission to publish this paper. They also wish to thank the Oldenbourg Verlag for permission to republish some matter which appeared in an earlier article.[5]

[4] These figures are calculated from Carter, *op. cit.*, (46) and (60).

[5] H. P. Schlaeppi and I. P. V. Carter, "Magnetic core memories using multiple coincidence," *Elektron. Rechenanlagen*, vol. 1, pp. 127-133; August, 1959. (In German.)

Magnetic Fields of Twistors Represented by Confocal Hollow Prolate Spheroids*

H. CHANG† AND A. G. MILNES‡, SENIOR MEMBER, IRE

Summary—A twistor is an anisotropic ferromagnetic cylindrical wire with nonmagnetic core. The intrinsic magnetization flux curls in helical sense in the wire and has an air return path. Many field problems must be solved for their successful use as information storage elements. For instance, the demagnetizing field in the wire causes instability of storage and therefore must be reduced by suitable geometry of the twistor. The flux lines emanating from a bit link neighboring windings and also impose a magnetic field intensity in neighboring bits. The interactions, although undesirable in packing bits in a memory array, can be used to advantage as operating forces in logical devices.[1]

This paper analyzes the demagnetizing field in a twistor bit, based on the geometrical model of a confocal hollow prolate spheroid and the magnetic characterization of the material by $\overline{B}=\mu_0(H+M)$ where $\overline{M}$ is the intrinsic magnetization, constant in magnitude, but oriented by the external field.

Demagnetizing factors for confocal hollow prolate spheroids are plotted against length-to-diameter ratio and wall thickness. Expressions for field intensities outside a twistor bit are given. Analogies between twistors and thin films are examined.

I. Introduction

TWISTORS refer to the class of ferromagnetic wires with helical direction of easy magnetization around the axis. The name was acquired in the early development[2] when twistors were made by physically twisting a piece of magnetostrictive wire to create a helical direction of maximum tension and a helical direction of maximum compression normal to the former [refer to Fig. 1(a)]. The easy direction of magnetization is in the direction of maximum tension or compression, depending on the material. Recent investigations have developed techniques such as electrodeposition[3] of Ni-Fe alloy on copper wire or tubing with helical direction of easy magnetization created during fabrication, or wrapping thin permalloy tapes around a wire in a helical direction.[4]

* Received by the PGEC, April 20, 1959; revised manuscript received, February 2, 1960. This investigation was carried out at Carnegie Inst. of Technology, and was supported in part by a Bell Telephone Labs. Fellowship (1957–1958) and in part by Office of Naval Res. Contract NONR 760(09).

† IBM Res. Lab., Poughkeepsie, N. Y.

‡ Electrical Engineering Dept., Carnegie Inst. of Technology, Pittsburgh, Pa.

[1] A. H. Bobeck and R. F. Fischer, "Reversible, diodeless, twistor shift register," *J. Appl. Phys.*, suppl. to vol. 30, pp. 43S–44S; April, 1959.

[2] A. H. Bobeck, "A new storage element suitable for large-sized memory arrays—the twistor," *Bell Sys. Tech. J*, vol. 36, pp. 1319–1340; November, 1957.

[3] S. J. Schwartz and J. S. Sallo, "Electrodeposited twistor and bit wire components," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. 8, pp. 465–469; December, 1959.

[4] K. Preston, Jr. and Q. W. Simkins, "Twistor buffer store," *Digest of Tech. Papers IRE-AIEE-Univ. of Pennsylvania Solid-State Circuits Conf.*, Philadelphia, Pa., pp. 14–15; February, 1959.

The helical magnetization path offers several attractive possibilities for application:

1) Cross-magnetizations in the axial and circumferential direction [see Figs. 1(a) and 2] allow control of the width of the magnetization vs field intensity curve in one direction by the presence of field in the other. Thus the twistor lends itself to logic operations.

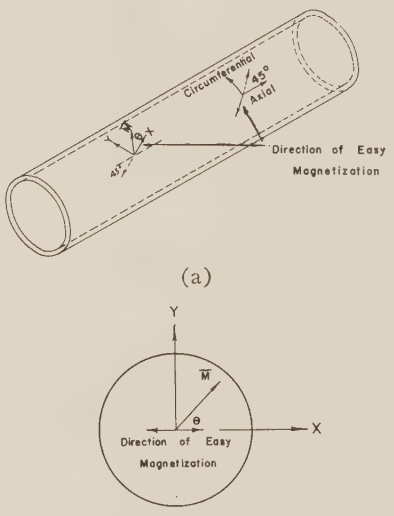


Fig. 1—Two-dimensional uniaxially anisotropic single magnetic domain representation of (a) hollow twistor and (b) thin film.

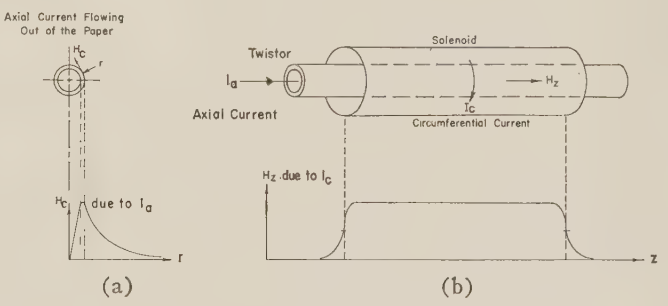


Fig. 2—Components of magnetic field in a hollow twistor bit. (a) Circumferential field. (b) Axial field.

2) The helical magnetization has a many-turned linkage with the magnetic wire as well as linkage with the coil around the wire. Hence, when the magnetic wire itself is used to sense the circumferential flux change, it is as effective in producing a voltage as a many-turned coil around the wire sensing the axial flux change.

The switching properties of both solid and hollow magnetic wires and schemes for using the twistors as memory devices have been studied by Bobeck.[2] Fabrication techniques and circuitry applications are reported in various conference papers.[3,4] This paper analyzes the field distributions of a twistor bit which is assumed to be single domained. Although a single-domained bit cannot always be realized, the analysis still gives the upper limit to the field intensity that may result from an actual bit.

In order for a magnetized bit to have stable storage, it is necessary to minimize the demagnetizing field which acts against the magnetization. Each helical magnetization flux line can be resolved into circumferential ($M \sin \theta$) and axial ($M \cos \theta$) components. The former circles the wire and closes on itself; therefore, it contributes no induced poles that can give rise to demagnetizing field. The axial component of magnetization has to leave the wire at one place or another to close its path through the air. At points of departure from the wire, magnetic poles are induced on the surface and cause a demagnetizing field opposing the axial magnetization. The magnitude of the demagnetizing field decreases as the length-over-diameter ratio of the magnetized bit increases, as can be inferred from the similar cases of rods or prolate spheroids whose demagnetizing fields have been analytically determined.[5] Another means to reduce the demagnetizing field is to use hollow wire; but the hollow wire carries less flux and, hence, yields a smaller switching voltage. In the following sections, the demagnetizing field and the demagnetizing factor in a hollow wire are determined analytically as a guide to optimum choice of geometrical dimension of a hollow twistor bit.

When twistor bits are packed in a memory array, it is important to space them so that no detrimental interference will occur between neighboring bits. However, with logic device applications, the interaction field intensity may be utilized as an operating force. Thus, study of the field distribution associated with each bit constitutes another objective of the paper.

II. Physical Model and Mathematical Analysis

The magnetic property is to be characterized mathematically by

$$\overline{B} = \mu_0(\overline{H} + \overline{M}) \tag{1}$$

where $\overline{B}$ = flux density, μ_0 = air permeability, $\overline{H}$ = field intensity and $\overline{M}$ = intrinsic magnetization.

The detailed implications of this characterization are discussed in the Appendix. The geometrical model for the solid twistor bit is chosen to be a prolate spheroid, while that for the hollow one is chosen to be a confocal hollow prolate spheroid. The potentials due to uniform magnetization in a ferromagnetic prolate spheroid can be derived[6] from the potential expressions for a prolate spheroid characterized magnetically by $\overline{B} = \mu \overline{H}$ and immersed in a field originally uniform and directed along the axis of rotation. Each total potential can be resolved into the sum of the original potential and a perturbation potential. The latter is attributed to induced magnetization which, incidentally, is uniform throughout the spheroid. Since the potential due to uniform magnetization M should be the same no matter whether it is intrinsic or induced, the perturbation potential due to induced magnetization is the solution we seek for a ferromagnetic spheroid.

For a hollow twistor bit, the natural geometrical model is a hollow circular cylinder of finite length. However, the mathematical difficulty associated with this geometry is prohibitive. Since a prolate spheroid with large major-to-minor-axes ratio a/b approximates a solid cylindrical wire, this naturally leads to the surmise that a hollow prolate spheroid bounded by two confocal spheroids of large major-to-minor-axes ratio approximates a hollow cylindrical wire. The spheroidal surfaces are chosen to be confocal for the mathematical reason that each bounding surface is specified by one value of a coordinate variable and, therefore, the matching of boundary conditions and the accompanying mathematical manipulations is simplified.

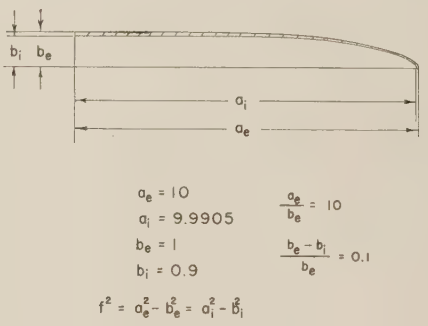


Fig. 3—Two confocal ellipses.

Fig. 3 depicts one quarter of an axial cross section of prolate spheroid, which approximates a hollow cylinder with length-to-diameter ratio of 10 and wall thickness 1/10 of the radius. As illustrated, the hollow spheroid has an almost uniform wall thickness and diameter equal to those of the hollow cylinder over a large portion in the middle, but these gradually diminish towards the ends and finally become vanishingly small at the ends. The wall thickness at the ends is only $(a_e - a_i)/(b_e - b_i) = 0.0095$ times that at the middle.

With larger major-to-minor-axes ratio, the approximation is even better and over a wide range of the ratio, wall thickness of the spheroid is always vanishingly

[5] R. M. Bozorth and D. M. Chapin, "Demagnetizing factors of rods," *J. Appl. Phys.*, vol. 13, p. 323; May, 1942. See Fig. 2.

[6] H. Chang, "Fields Associated with Ellipsoids—with Applications to Shielding, Thin Films and Twistors," Ph.D. dissertation, Carnegie Inst. Tech., Pittsburgh, Pa., Naval Res. Rept. NONR 760(09), pp. 88–90; March, 1959.

mall at the ends. Thus, a hollow spheroid bounded by wo confocal prolate spheroidal surfaces represents easonably well a hollow cylinder with open ends.

With the geometrical model chosen, Section III obains the mathematical solution, and Section IV proceeds to examine the field distribution in a hollow prote spheroid with $\overline{B}=\mu\overline{H}$ and immersed in a uniform eld. After finding that the resulting field intensity is ssentially uniform both in direction and in magnitude n the spheroid, it is justified to define demagnetizing eld ($H_d=H-H_0$), induced magnetization

$$\left(M=\frac{\mu-\mu_0}{\mu_0}H\right),$$

nd demagnetizing factor (demagnetizing field per unit nagnetization). Computation illustrates that demagetizing factor is nearly independent of the permeability μ) and thus, by its very definition, the demagnetizing actor should equally apply to a hollow prolate spheroid vith uniform intrinsic magnetization (which approximately represents a hollow twistor bit with only an axial nagnetization component). The above outline is to be laborated on in Sections IV and V.

III. Fields Associated with Hollow Prolate Spheroid

Consider a hollow prolate spheroid (Fig. 4) of contant permeability μ, bounded by two confocal sphe-oidal surfaces η_i and η_e, and embedded in free space μ_0 *i.e.*, $\mu_e=\mu_i=\mu_0$). Before the insertion of the spheroid, uniform field H_{0x} was directed along the X axis. The olution for the boundary-value problem is obtained by

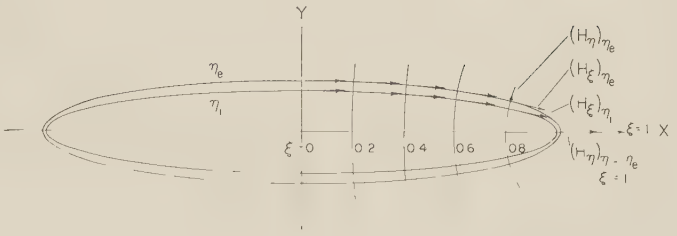


Fig. 4—Field intensities in a hollow prolate spheroid immersed in an originally uniform field along X axis (Note that $\overline{H}\cong H_\xi\bar{\imath}_\xi\cong H_\xi\bar{\imath}_x$).

Chang.[7] The potential of the applied field is

$$\begin{aligned} P_0 &= -H_{0x}X \\ &= -H_{0x}f\xi\eta \\ &= -H_{0x}fP_1(\xi)P_1(\eta) \end{aligned} \tag{2}$$

where H_{0x}=original uniform field intensity in the x direction, $f=\sqrt{a^2-b^2}$, a, b=major, minor axes of generating ellipse, ξ, η=prolate spheroidal coordinates. The presence of the spheroid of permeability μ disturbs the original uniform field, and the disturbed potentials for the three regions separated by η_i and η_e are given below:

$$P_e = AP_1(\xi)[P_1(\eta)+BQ_1(\eta)], \tag{3}$$

$$P = CP_1(\xi)[P_1(\eta)+DQ_1(\eta)], \tag{4}$$

$$P_i = EP_1(\xi)P_1(\eta), \tag{5}$$

where

$$P_1(\xi)=\xi, \qquad P_1(\eta)=\eta,$$

and

$$Q_1(\eta)=\frac{\eta}{2}\ln\frac{\eta+1}{\eta-1}-1, \tag{6}$$

$$A=-fH_{0x}, \tag{7}$$

$$D=\left[\frac{(\mu-\mu_0)P_1(\eta)\dfrac{d}{d\eta}Q_1(\eta)}{\mu_0Q_1(\eta)\dfrac{d}{d\eta}P_1(\eta)-\mu P_1(\eta)\dfrac{d}{d\eta}Q_1(\eta)}\right]_{\eta_i} \tag{8}$$

$$C=\left\{\frac{\mu_0A\left[Q_1(\eta)\dfrac{d}{d\eta}P_1(\eta)-P_1(\eta)\dfrac{d}{d\eta}Q_1(\eta)\right]}{\mu Q_1(\eta)\dfrac{d}{d\eta}P_1(\eta)-\mu_0P_1(\eta)\dfrac{d}{d\eta}Q_1(\eta)+D(\mu-\mu_0)Q_1(\eta)\dfrac{d}{d\eta}Q_1(\eta)}\right\}_{\eta_e}, \tag{9}$$

$$B=\left(\frac{C}{A}-1\right)\frac{P_1(\eta_e)}{Q_1(\eta_e)}+\frac{CD}{A}, \tag{10}$$

$$E=C\left[1+D\frac{Q_1(\eta_i)}{P_1(\eta_i)}\right]. \tag{11}$$

[7] *Ibid.*, ch. 5, sect. 2.3.

Field Intensities in Hollow Spheroid

The field intensity in prolate spheroidal coordinates is:

$$\begin{aligned}\overline{H} &= -\overline{\nabla} P \\ &= H_\xi \bar{i}_\xi + H_\eta \bar{i}_\eta \\ &= -\frac{1}{h_\xi}\frac{\partial P}{\partial \xi}\bar{i}_\xi - \frac{1}{h_\eta}\frac{\partial P}{\partial \eta}\bar{i}_\eta, \end{aligned} \tag{12}$$

where

$$h_\xi = f\sqrt{\frac{\eta^2 - \xi^2}{1 - \xi^2}}, \tag{13}$$

$$h_\eta = f\sqrt{\frac{\eta^2 - \xi^2}{\eta^2 - 1}}. \tag{14}$$

The first term H_ξ is the component of the field intensity tangential to a prolate spheroidal surface η, and the second term H_η is the normal component (refer to Fig. 4). Notice that the θ component is absent, since P is independent of θ. Thus, it is found that

$$H_\xi = -\frac{1}{f}\sqrt{\frac{1 - \xi^2}{\eta^2 - \xi^2}}\, C[P_1(\eta) + DQ_1(\eta)], \tag{15}$$

$$H_\eta = -\frac{1}{f}\sqrt{\frac{\eta^2 - 1}{\eta^2 - \xi^2}}\, CP_1(\xi)\left[1 + D\frac{d}{d\eta}Q_1(\eta)\right]. \tag{16}$$

The flux density is related to the field intensity by $\overline{B} = \mu\overline{H}$.

Flux Density in a Spheroid with Infinite Permeability

When the permeability μ approaches infinity, although the flux density in the spheroid will increase, yet the field intensity will diminish since the body is approaching equipotential. In the limit, the flux density reaches a finite value as determined by the series air path, and the field intensity becomes zero. But for the passage of the flux through the air, the flux density would become infinite as permeability approaches infinity.

The limiting case with μ infinite has simpler expressions than those for μ large but finite, and therefore serves as a good approximation for the calculation of some quantities. The coefficients for this limiting case, from (8) and (9), are given below:

$$\lim_{\mu\to\infty} D = -\left[\frac{\frac{d}{d\eta}P_1(\eta)}{\frac{d}{d\eta}Q_1(\eta)}\right]_{\eta_i} = -\frac{1}{\left[\frac{d}{d\eta}Q_1(\eta)\right]_{\eta_i}}, \tag{17}$$

$$\lim_{\mu\to\infty} \mu C = \left\{\frac{\mu_0 A\left[Q_1(\eta)\frac{d}{d\eta}P_1(\eta) - P_1(\eta)\frac{d}{d\eta}Q_1(\eta)\right]}{Q_1(\eta)\frac{d}{d\eta}P_1(\eta) - \frac{1}{\left[\frac{d}{d\eta}Q_1(\eta)\right]_{\eta_i}}Q_1(\eta)\frac{d}{d\eta}Q_1(\eta)}\right\}_{\eta_e}. \tag{18}$$

IV. Computation and Interpretation of Results

The field distributions, in terms of

$$\frac{H_\xi}{H_{0x}}, \quad \frac{H_\eta}{H_{0x}}, \quad \frac{B_\xi}{H_{0x}}, \quad \frac{B_\eta}{H_{0x}},$$

as derived in the preceding section, have been computed over the following wide ranges of parameter values on an IBM 650 computer:

$\frac{a_e}{b_e}$ (Major-to-minor-axis ratio)	2 to 1000
$\frac{b_i}{b_e}$ (Ratio of internal to external radius, a measure of the wall thickness)	0 to 0.99
$\frac{\mu}{\mu_0}$ (Relative permeability)	100 to infinity.

Bulky data result from the above computation. Fortunately, it is found that, except near the two ends of the spheroid, the field intensity is almost independent of ξ and η, and is approximately equal to the tangential field intensity H_ξ. Fig. 5 exemplifies the worst situation, but, even then, the field intensity still appears substantially independent of ξ and η. Essential features of the results are itemized below:

1) Since

$$\eta = \left[1 - \frac{1}{\left(\frac{a}{b}\right)^2}\right]^{-1/2}, \tag{19}$$

a hollow prolate spheroid, then with major-to-minor-is ratio for the external bounding surface to be $a_e/b_e \geq 5$, the value of η is always approximately 1 roughout the spheroid.

As

$$X = f\xi\eta = \sqrt{a^2 - b^2}\xi\eta, \tag{20}$$

en

$$X \cong a\xi, \tag{21}$$

r

$$a \gg b \quad \text{and} \quad \eta \cong 1.$$

oreover, as illustrated by Fig. 4 (for $a_e/b_e = 5$, and $/b_e = 0.8$), over the major portion of the spheroid

$$\bar{\imath}_\xi \cong \bar{\imath}_X \tag{22}$$

us,

$$\frac{X}{a}\bar{\imath}_x = \xi\bar{\imath}_\xi \tag{23}$$

d with increase in the major-to-minor-axis ratio, the pproximation improves.

2) The component of field intensity tangential to a heroidal surface H_ξ has nearly the same magnitude er the ranges of η from η_i to η_e and ξ from -0.8 to 0.8, seen for the worst case computed, for $\mu_r = 10^2$ and $/b_e = 2$ in Fig. 5. This uniformity in magnitude im-roves greatly as μ_r or a_e/b_e increases.

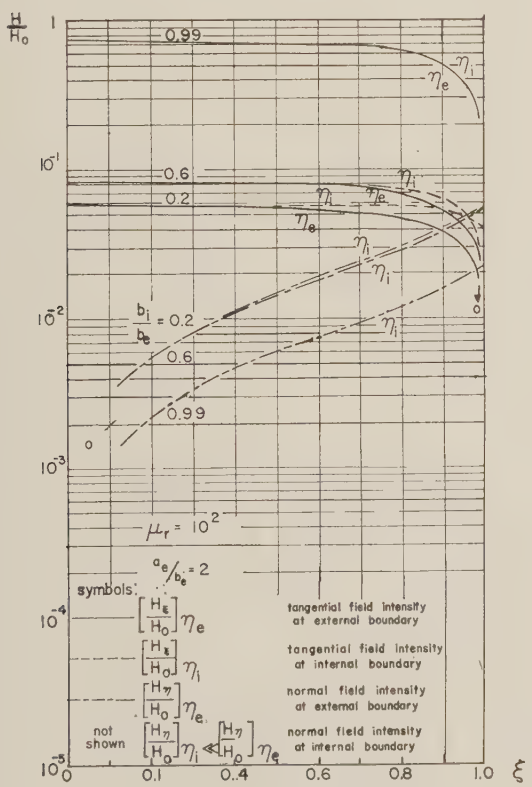


Fig. 5—Plot of field intensities.

3) The component of field intensity normal to a spheroidal surface H_η is zero in the middle of the spheroid where $\xi = 0$, and increases towards the ends. Over the major portion of the spheroid, *e.g.*, $\xi = -0.8$ to 0.8, H_η is less than H_ξ by orders of magnitude,

$$H_\eta \ll H_\xi. \tag{24}$$

Although it is not shown in the figures, the computed results indicate also that

$$(H_\eta)_{\eta_i} \ll (H_\eta)_{\eta_e} \ll H_\xi. \tag{25}$$

4) In the light of (22) and (24), we have

$$\begin{aligned} \bar{H} &= H_\xi\bar{\imath}_\xi + H_\eta\bar{\imath}_\eta \\ &\cong H_\xi\bar{\imath}_\xi \\ &\cong H_\xi\bar{\imath}_x. \end{aligned} \tag{26}$$

Hence the field intensity is almost uniform in direction as well as in magnitude in the hollow spheroid for ξ from -0.8 to 0.8 and η from η_i to η_e.

This result has two very important consequences:

a) The disturbed field intensity in the spheroid with constant μ can be described by a single tangential field component H_ξ, *e.g.*, $\xi = 0$ and $\eta = \eta_e$. Fig. 6 shows H_ξ/H_0 as a function of a_e/b_e, b_i/b_e for $\mu_r = 10^4$.

b) The uniformity of field intensity over a major portion of the hollow spheroid results in uniform induced magnetization, since

$$\bar{M} = \frac{\mu - \mu_0}{\mu_0}\bar{H}.$$

Thus, demagnetizing factors (defined as demagnetizing field per unit magnetization) are meaningful for hollow spheroids with constant μ.

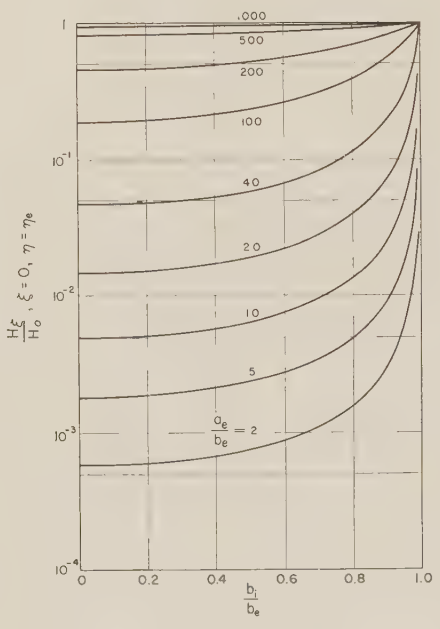


Fig. 6—Tangential field intensity, $\mu_r = 10^4$.

V. Demagnetizing Factors for Hollow Prolate Spheroids

In the preceding section, it has been established that when a hollow prolate spheroid is immersed in an originally uniform field along its long axis, the disturbed field intensity inside the material with permeability μ is essentially uniform both in direction and magnitude. Since both the original and the disturbed field intensities are uniform, the perturbation field intensity, as the difference between the two , is also uniform. Furthermore, the induced magnetization, as defined by

$$M = \frac{\mu - \mu_0}{\mu_0} H, \tag{27}$$

is also uniform as a consequence of uniform disturbed field intensity H. Thus we may define

$$H = H_0 + H_d = H_0 - NM, \tag{28}$$

where

H = disturbed field intensity,
H_0 = original field intensity,
H_d = demagnetizing field intensity,
N = demagnetizing factor (or demagnetizing field intensity per unit magnetization),
$M = (\mu - \mu_0)/(\mu_0)H$, induced magnetization.

Notice that algebraic rather than vector notation is used since all vector quantities involved are either parallel or antiparallel. Eqs. (27) and (28), when combined to eliminate M, give

$$N = \frac{1 - \dfrac{H}{H_0}}{(\mu_r - 1)\dfrac{H}{H_0}}, \tag{29}$$

where

$$\mu_r = \frac{\mu}{\mu_0}.$$

For a hollow prolate spheroid (with constant permeability μ), it has been shown that

$$\overline{H} \cong H_\xi \bar{\imath}_\xi \cong H_\xi \bar{\imath}_x. \tag{30}$$

When H is substituted by H_ξ at $\xi = 0$, and $\eta = \eta_e$, (29) becomes

$$N = \frac{1 - \dfrac{(H_\xi)_{\xi=0,\eta=\eta_e}}{H_0}}{(\mu_r - 1)\dfrac{(H_\xi)_{\xi=0,\eta=\eta_e}}{H_0}}, \tag{31}$$

where

$$(H_\xi)_{\xi=0,\eta=\eta_e} = -\frac{1}{f}\sqrt{\frac{1}{\eta_e^2 - 1}}\, C[P_1(\eta_e) + DQ_1(\eta_e)]. \tag{32}$$

The above analytical expression for the demagnetizing factor still involves μ_r in the constants C and D. However, when calculations are carried out, it is found that up to three significant figures, the demagnetizing factor is independent of permeability in the range $\mu_r = 10^2$ to ∞. Consequently, in the light of (28), the demagnetizing factor found above should also apply to a ferromagnetic hollow prolate spheroid with uniform intrinsic (instead of induced) magnetization along the major axis.

The above results are not surprising for the following reasons. The demagnetizing field is due to the magnetic poles formed on the boundary between the magnetic material (μ) and the ambient (μ_0). Since there are relatively few flux lines entering the enclosed air space, the magnetic poles formed on the inner bounding surface are of weaker strength and have comparatively negligible influence on the demagnetizing field. So we need only consider the magnetic poles on the external boundary of the spheroid. The magnetic pole distribution and strength on the external surface is determined by the magnetization distribution in the spheroid, or, rather by how the magnetization flux lines leave the spheroid. It has been shown that the induced magnetization in the case of a solid spheroid is exactly uniform and parallel to the major axis if the original field is along the major axis. It has also been shown that the induced magnetization in the case of a hollow spheroid is approximately uniform and parallel to the major axis. Hence, in both cases, the flux lines leave the external boundary in the same fashion and the induced magnetic pole distribution should be of the same pattern. As it is known that, for a solid spheroid, the demagnetizing factor is independent of permeability or magnetization, the demagnetizing factor of a hollow prolate spheroid with identical pole distribution should also be independent of permeability or magnetization.

Consider solid and hollow prolate spheroids of the same external major-to-minor-axes ratio. Although they have identical pole distributions on the external surface, the poles are of different strength since the solid spheroid emanates $\mu_0 M\pi b_e^2$ flux lines from its right half while the hollow spheroid only emanates $\mu_0 M\pi(b_e^2 - b_i^2)$ flux lines. The demagnetizing field for the hollow case is therefore only

$$\frac{\mu_0 M\pi(b_e^2 - b_i^2)}{\mu_0 M\pi b_e^2} = 1 - \left(\frac{b_i}{b_e}\right)^2 \tag{33}$$

that of the solid case. Since both cases have the same magnetization, the demagnetizing factor of the hollow prolate spheroid is $[1 - (b_i/b_e)^2]$ that of the solid one.

VI. Field Distribution Outside a Twistor Bit

The field distribution outside a solid twistor bit is derived by Chang,[8] and that outside a hollow twistor bit with inner-to-external-radii ratio b_i/b_e can be approx

[8] *Ibid.*, p. 91.

ately calculated by using $M_x[1-(b_i/b_e)^2]$ instead of in the expressions for a solid bit, which follow:

$$\frac{H_x}{M_x\frac{ab^2}{f^3}} = \frac{1}{2}\ln\frac{\eta+1}{\eta-1} - \frac{\eta}{\eta^2-\xi^2},$$

$$\frac{H_y}{M_x\frac{ab^2}{f^3}} = \frac{\xi(1-\xi^2)^{1/2}}{(\eta^2-1)^{1/2}(\xi^2-\eta^2)}\cos\theta,$$

$$\frac{H_z}{M_x\frac{ab^2}{f^3}} = \frac{\xi(1-\xi^2)^{1/2}}{(\eta^2-1)^{1/2}(\xi^2-\eta^2)}\sin\theta, \qquad (34)$$

ere the prolate spheroidal coordinates (η, ξ, θ) are ated to the rectangular coordinates (X, Y, Z) by

$$\frac{X^2}{\eta^2} + \frac{Y^2+Z^2}{\eta^2-1} = f^2,$$

$$\frac{X^2}{\xi^2} - \frac{Y^2+Z^2}{1-\xi^2} = f^2,$$

$$\theta = \tan^{-1}\frac{Z}{Y}, \qquad (35)$$

d

M_x = axial component of magnetization,
a = major axis of external spheroid,
b = minor axis of external spheroid,
$f = \sqrt{a^2-b^2}$.

VII. Conclusions

1) To immerse a hollow prolate spheroid with permeility μ in an originally uniform field along its major is is found to result in approximately uniform field d uniform induced magnetization along the major is in the magnetic material. The variation of this field a function of major-to-minor-axis ratio and wall ickness is exemplified by Fig. 6.

2) The uniformity of field and magnetization enables e definition of the demagnetizing factor N as the deagnetizing field per unit magnetization. For a linear $\overline{B}=\mu\overline{H}$) hollow prolate spheroid, N is independent of rmeability ($\mu_r=10^2$ to ∞). Hence, in the light of (28), should apply to a ferromagnetic $[\overline{B}=\mu_0(\overline{H}+\overline{M})]$ llow spheroid as well. The demagnetizing factor as a nction of major-to-minor-axes ratio and wall thickss is plotted in Fig. 7.

The demagnetizing field is the product of the demagtizing factor and the magnetization which can be und by knowing the intrinsic magnetization or the duced magnetization

$$\left(M = \frac{\mu-\mu_0}{\mu_0}H\right).$$

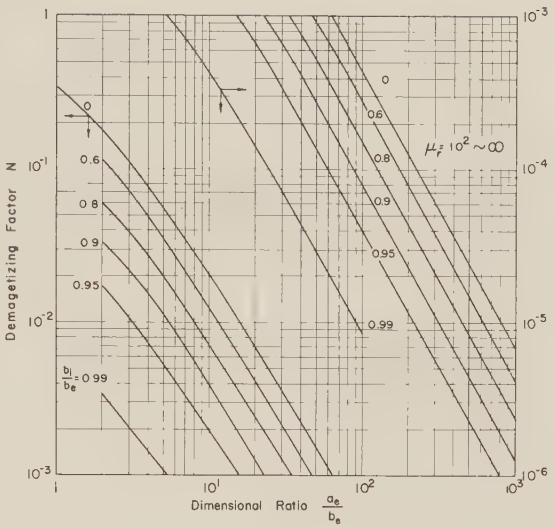


Fig. 7—Demagnetizing factors (or demagnetizing field intensity per unit magnetization) of prolate spheroids, solid or hollow, $\mu_r=10^2$ to ∞.

For values of parameter b_i/b_e not shown in Fig. 7, interpolation can be readily performed.

Since the demagnetizing factor is independent of relative permeability, a simplified form of (31) can be employed for computation:

$$\lim_{\mu_r\to\infty} N = \frac{1-\lim_{\mu_r\to\infty}\frac{H_\xi}{H_0}}{\lim_{\mu_r\to\infty}\frac{\mu H_\xi}{H_0} - \lim_{\mu_r\to\infty}\frac{H_\xi}{H_0}} = \frac{1}{\lim_{\mu_r\to\infty}\frac{\mu H_\xi}{H_0}}. \qquad (37)$$

3) The ratio of demagnetizing factors for solid and hollow prolate spheroids of the same external bounding surface is approximately equal to the ratio of their cross-sectional areas in the middle (see Section V):

$$\frac{N_{\text{hollow}}}{N_{\text{solid}}} = \frac{\pi(b_e^2-b_i^2)}{\pi b_e^2} = 1-\left(\frac{b_i}{b_e}\right)^2. \qquad (38)$$

4) When the above results are applied to a twistor, the longitudinal component of the helical magnetization is the magnetization along the major axis and is used in all the above computations. The circumferential component of the magnetization closes on itself and causes no induced poles and therefore no demagnetizing field. After choosing the bit length ($2a_e$) on a magnetic wire with known I.D. ($2b_i$), O.D. ($2b_e$), and magnetization M, Fig. 7 readily yields the value of the demagnetizing factor N and hence demagnetizing force ($-NM\cos\theta$). A small demagnetizing force is essential for stable magnetic state and squareness of B-H loop when the switching mechanism is wall motion.

5) The field distribution outside a twistor can be computed by using (34) and (35). As mentioned in the Summary, such information should prove useful in evaluating interaction between bits.

Appendix

Magnetic Hysteresis in Twistors Derived from Energy Considerations

A single domain model[9–11] with one or many easy axes of magnetization, successfully explains M-H loops resulting from coherent rotation in thin films. With the advent of electrodeposition technique, hollow twistors with very thin magnetic shells are feasible and behavior comparable to that of thin films is quite possible. In this Appendix, M-H loops for hollow twistors, as would result from coherent rotation, are derived, thus providing a basis for experimental verification of the switching mechanism in thin hollow twistors. Analogies between thin films and twistors are made clear in this treatment to indicate that device applications with one may suggest possibilities for the other.

Conditions assumed for the derivation of critical curves and M-H loops are:

1) The magnetization is uniform in magnitude and direction throughout the magnetic domain. Its magnitude M is assumed to be independent of H, but with a direction changeable by an applied magnetic field of adequate strength.

2) The magnetization is confined to a two-dimensional coordinate space such that the demagnetizing field will be small and the magnetic state will be stable. For a thin film, it is an x-y-coordinate space [Fig. 1(b)]. For a hollow twistor bit, it is a cylindrical surface [Fig. 1(a)].

3) Uniform field is applied in the plane of magnetization. In the case of a thin film, this can be easily realized. In the case of a hollow twistor bit, a current flowing in the nonmagnetic core will create an essentially uniform circumferential field in the thin magnetic sheet while a solenoid around the wire with length much larger than its diameter will provide a uniform axial magnetic field inside the solenoid, and a rapidly diminishing field outside the solenoid (Fig. 2).

4) One easy axis of magnetization is assumed. For a thin film, this axis may be taken as the X axis. For a twistor bit, the easy axis is in the direction of a helical line at an angle θ (usually 45°) with the axis of the cylinder. If we imagine the cylindrical surface rolled out into a plane, the helical line becomes a straight line and it is designated as the X direction, and the perpendicular direction as the Y direction.

The energy per unit volume of a single magnetic domain is

$$E = -\mu_0 \overline{H} \cdot \overline{M} + K \sin^2 \theta$$
$$= -\mu_0 M (H_x \cos\theta + H_y \sin\theta) + K \sin^2 \theta, \quad (39$$

where

$-\mu_0 \overline{H} \cdot \overline{M}$ = magnetization energy (joule/meter^3),
$K \sin^2 \theta$ = anisotropy energy (joule/meter^3),
μ_0 = permeability of free space (henry/meter)
$\overline{H}$ = field intensity (ampere turn/meter),
$\overline{M}$ = intrinsic magnetization (ampere turn meter),
K = anisotropic constant (joule/meter^3),
θ = angle between magnetization and the easy axis of magnetization [refer to Fig. 1(a)]

Figs. 8 and 9(a) (next page) are derived by Slonczewski using the above energy relation.[9] The straight lines in Fig. 8 are equilibrium lines corresponding to conditions of minimum energy; *viz.*,

$$\frac{\partial E}{\partial \theta} = 0,$$

and

$$\frac{\partial^2 E}{\partial \theta^2} > 0.$$

From (39)

$$\frac{\partial E}{\partial \theta} = \mu_0 M H_x \sin\theta - \mu_0 M H_y \cos\theta$$
$$+ 2K \sin\theta \cos\theta = 0. \quad (40$$

The slope of an equilibrium line, from (40), is found to be $\partial H_y / \partial H_x = \tan\theta$, which coincides with the direction of the magnetization vector. Thus each point on a line not only indicates the field intensity components H_x and H_y by its coordinates, but also the direction of magnetization by the inclination of the line.

The critical curve is determined from the conditions

$$\frac{\partial^2 E}{\partial \theta^2} = \frac{\partial E}{\partial \theta} = 0, \quad (41$$

which divides the line

$$\frac{\partial E}{\partial \theta} = 0$$

into two segments; *viz.*,

$$\frac{\partial E}{\partial \theta} = 0, \quad \frac{\partial^2 E}{\partial \theta^2} > 0 \text{ (minimum energy equilibrium line),}$$

and

$$\frac{\partial E}{\partial \theta} = 0, \quad \frac{\partial^2 E}{\partial \theta^2} < 0 \text{ (maximum energy, not shown in Fig. 8)}$$

[9] J. C. Slonczewski, "Theory of Magnetic Hysteresis in Films and Its Applications to Computers," IBM Res. Lab., Poughkeepsie, N. Y., Res. Memo. RM003.111.224; October, 1956.

[10] E. M. Bradley and M. Prutton, "Magnetization reversal by rotation and wall motion in thin films of nickel-iron alloys," *J. Electronics Control (London)*, vol. 6, pp. 81–96; January, 1959.

[11] R. E. Behringer, "Theory of Critical Curves and Hysteresis Loops in Thin Magnetic Films," IBM Res. Lab., Poughkeepsie, N. Y., Res. Memo. RC 63 (unpublished); October, 1958.

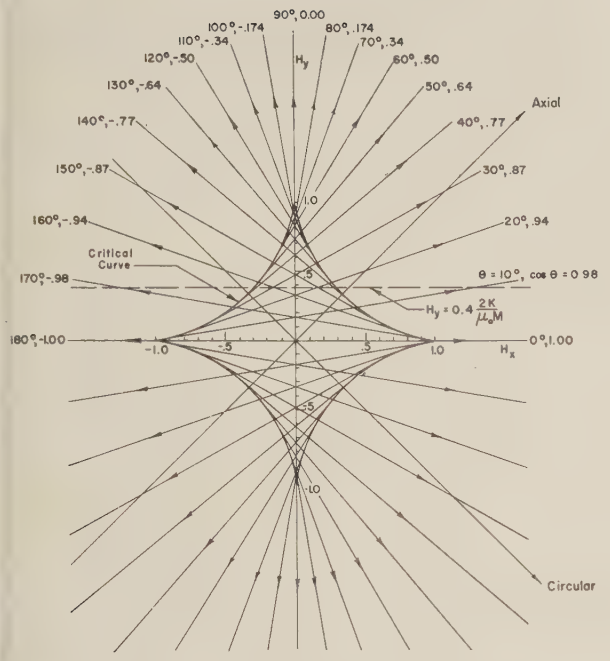


Fig. 8—The orientation of M, indicated by arrows, depends on H. H_y and H_x are in units of $2K/\mu_0 M$ (after Slonczewski[9]).

rom (39) and (40), the critical curve is found as

$$H_x^{2/3} + H_y^{2/3} = \left(\frac{2K}{\mu_0 M}\right)^{2/3}. \qquad (42)$$

Within the critical curve, each point, through which wo equilibrium lines pass, corresponds to two equibrium states. Outside the critical curve, each point corresponds to one equilibrium state. Thus in M-H loops, as etermined from the equilibrium lines, for field intensies in the range $\pm H_c$, coercive force (corresponding to he region within the critical curve), there are two possile values for the magnetization, while for values of eld outside the range of $\pm H_c$ there is only one value for nagnetization. The curves as shown in Fig. 9 are derived rom Fig. 8. It is interesting to note that dc B-H shapes

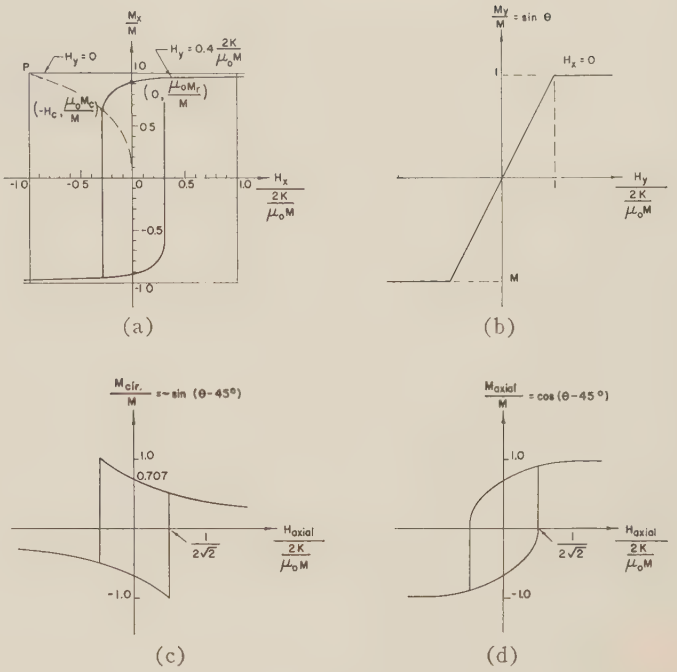


Fig. 9—Hysteresis loops or thin films and twistors (derived from energy considerations for a single domain). (a) M_x vs H_x (H_y as parameter). Hysteresis loop of a thin film in easy direction. (b) M_y vs H_y ($H_x = 0$). Hysteresis loop of a thin film in hard direction. (c) $M_{cir.}$ vs H_{axial} ($H_{cir.} = 0$) for a hollow twistor. (d) M_{axial} vs H_{axial} ($H_{cir.} = 0$) for a hollow twistor.

similar to Fig. 9(c) and (d) have been observed experimentally for twistors, and have also been reported for thin films.[10]

Acknowledgment

Many stimulating discussions with A. H. Bobeck at Bell Telephone Laboratory, Murray Hill, N. J., during the summer employment of the first author in 1957 led to the present work. The computation was carried out by Mrs. E. Aveman at BTL. The authors wish to express their gratitude to both these persons.

The Design of a General-Purpose Microprogam-Controlled Computer with Elementary Structure*

THOMAS W. KAMPE†

Summary—This paper presents the design of a parallel digital computer utilizing a 20-μsec core memory and a diode storage microprogram unit. The machine is intended as an on-line controller and is organized for ease of maintenance.

A word length of 19 bits provides 31 orders referring to memory locations. Fourteen bits are used for addressing, 12 for base address, one for index control, and one for indirect addressing. A 32nd order permits the address bits to be decoded to generate special functions which require no address.

The logic of the machine is resistor-transistor; the arithmetic unit is a bus structure which permits many variants of order structure.

In order to make logical decisions, a "general-purpose" logic unit has been incorporated so that the microcoder has as much freedom in this area as in the arithmetic unit.

Introduction

THIS PAPER discusses the logical design of a binary, parallel, real-time computer. Only those aspects of packaging and circuitry which bear directly on this topic will be considered.

Since the specifications for the job a computer is to perform are not enough to fix the design, the logical designer is faced with an undetermined system. One of his main functions is to analyze the system in its natural environment, *i.e.*, with malfunctions, operator errors, etc., and to supply the remainder of the side conditions which do fix the design.

In this discussion, the exposition will be directed toward the design philosophy which led to a machine now being built. In order to accomplish this, we shall consider the functional requirements, their analysis in terms of the state of the art, the basic design decisions, and, finally, a description of the computer as it stands.

Functional Requirements

The design of the computer (known, for a variety of reasons, as the SD-2) was undertaken to supply a computer capable of moderately fast arithmetic with perhaps five decimal places of accuracy and 3000 or more words of storage. Furthermore, the computer must reside in a hostile environment (a small house, 0° to 85°C temperature), withstand severe shocks, and be maintained by men with only two weeks training on the system. The volume limitation is 40 cubic feet. Within this space must reside the control computer, memory, power supplies, complete maintenance facilities, and sufficient input/output equipment to handle 20 shaft position outputs, 30 such inputs, numerous switch settings, and 20 or more display or relay signals.

The final specification (or blow) was that 15 months were available from the start of preliminary design to the delivery of an operating instrument with debugged program.

Design Analysis

The maintenance requirement was evidently the major problem. In order to achieve the simplicity required, two design criteria were necessary.

First, the computer had to be readily understood. This implied that the usual clever logical tricks such as intensive time sharing of control and arithmetic were undesirable.

Second, if built-in maintenance facilities were to be kept simple, the machine must be designed with this in mind.

Since temperature and reliability were important, an extremely conservative approach had to be taken with respect to component performance.

With the schedule requirements, a machine which could be designed and released in pieces was needed. Since the control system is usually the most troublesome part of a computer to design, a simple control was needed.

The volume available, together with the schedule, required a logical design with natural packaging properties in the sense that it should break, in a natural way, into logical packages of a reasonable size having a minimum of interpackage communication.

Design Decisions

The need for 2000 operations per second poses a serious access problem with a serial memory, unless one resorts to several simultaneously operating control units which are neither small nor simple. Hence, a random access memory seemed advisable. Magnetic core memories at 85°C are a problem, but they can be built, provided memory cycle time is not too short. The memory was chosen as 4096 words of core storage, with a 20 μsec cycle time.

The requirement for training a man in two weeks to maintain the machine argues for a simple-structured parallel machine. Providing that much use is made of asynchronous transfer, there are a variety of simple maintenance methods, particularly if a bus structure is adopted. Also, asynchronous, or semi-asynchronous parallel machines require only average performance of

* Received by the PGEC, September 28, 1959; revised manuscript received, March 6, 1960. This paper was presented at the Sixth Annual University of Denver Computer Symposium, Denver, Colo., July, 1959, and subsequently printed for limited distribution by Librascope, Glendale, Calif.

† Advanced Projects Group, Shipboard Equipment Dept., Librascope, Glendale, Calif.

of components, not of any particular component; the ntral limit theorem of statistics can come to the aid of liability. This approach was finally adopted.

The simplicity of both design and understanding is led by the use of a microprogram control system. rther, maintenance is made rather simple by two pro-sions on the maintenance console.

The first of these is a manner of going through the croprogram on a step-by-step basis. While this tests tle of the dynamics, it can often locate totally de-tive parts, and it helps factory checkout immeasur-ly.

The second is a means of taking out the microprogram it and substituting a set of switches. This permits a aintenance man to exercise specific registers, or the emory, at will.

This is a powerful tool, and is almost free with a icroprogram control. Finally, and rather pragmat-ally, microprogramming permits "last minute" changes machine operation without serious hardware modi-ations. This approach was chosen.

Regardless of the control used, at various times in the ocess of executing orders, decisions must be made. ccasionally these are on a single bit, more often on vo, and occasionally on more than two. If one excludes der decoding, only such functions as zero detection quire the use of more than two bits. At this point, the gical designer is faced with a rather sticky decision: hether to design a specific set of decision logic, which cheap to build but sometimes messy, or to use some icrocontrolled logic-generating scheme.

In this case, the latter alternative was taken. A unit, lled (for several obscure reasons) the alteration unit as designed which amounted to a three-address, one-it unit. It can generate any Boolean function of two inary variables and transmit this value to another vari-ble. A special set of logic was needed for detecting eros.

Because of the rather wild nature of the inputs, it eemed desirable to include a trapping mode. The logic or this was made an adjunct to the alteration unit.

The circuitry chosen was resistor-transistor logic, hich yields either Sheffer stroke or NOR logic, as one refers, high or low true logic, and *p-n-p* or *n-p-n* tran-istors. In this case, the combination was high true logic nd *p-n-p* transistors, so that the logical operation is heffer stroke. Because of temperature and reliability equirements, the maximum frequency available was a 50-kc square wave. This gave a cycle time of 4 μsec vailable for asynchronous transfer in any sequence of gic.

An index register seemed advisable because of the mount of data processing. Thus, additions were needed or indexing, arithmetic, and counter advance. It eemed undesirable to have more than one parallel dder, so that an adder accessible to all registers was hosen. This was another argument for a bus structure.

Because of the multiplicity of problems being handled multaneously, one index register was not really enough. Rather than add another register, indirect addressing was chosen.

At this point, one needs 12 bits for address, one for index tagging, and one to specify whether the address is direct or indirect, or 14 bits for operand selection. Thirty-two orders was a tight minimum, so the minimum word length was 19 bits. Since this was consistent with five decimal place accuracy, it was tentatively chosen. It was decided, however, to design a structure basically suited to any length word.

Shifting is necessary to multiply and divide and is required on two registers, yet shift registers for asynchronous operation are complex. Hence, it was decided to put the shift facility on the data transfer bus. By providing complementing here, subtraction could be generated.

It was decided to use two-complement arithmetic, first because of the simplicity of the multiply-divide logic, and second because it avoids the whole negative zero question.

The precise number of microsteps needed was determined by a trial microprogram. The machine was designed for up to 512 microsteps although only 384 are now used. Eight bits were in a register, called *J*, and one was a flip-flop, *TO*, in the alteration unit, thus allowing fixed sequence with a one-bit microprogrammed choice. This, incidentally, is the genesis of the name "alteration unit."

The SD-2 Computer

Fig. 1 is a block diagram of the computer. There will be, presently, a block-by-block description of the computer.

The two boxes on the left were added to facilitate input and output. The output buffer holds 20 words, and outputs all values in a 4.8-msec cycle, thus providing for nearly continuous outputs. The output distributor is a selection system which allows the programmer to transmit the contents of the accumulator onto one of eight channels to control external devices. The "inputs" line represents up to 32 channels which can be read into the accumulator. The numbers 8 and 32 are purely

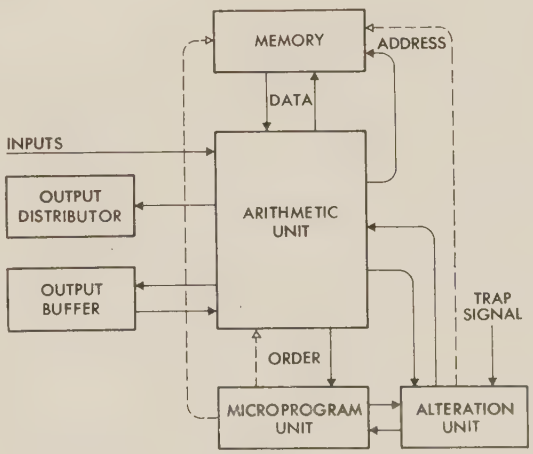


Fig. 1—Computer block diagram.

arbitrary; the upper limit of 32 is a microcode convenience only.

The alteration unit, in addition to its decision making duties, has several other functions. It has a five bit counter, used for microsubroutines, which can be set to any value chosen or to any number on the arithmetic unit. The alteration unit can sense when it goes from all zeros to all ones. In addition, the flip-flops controlling initial carry in the adder, end carry in shifting, and memory read or write control are in this unit.

Fig. 2 is a block diagram of the arithmetic unit. Information may be put onto the *b* bus from any register, or from outside sources, such as inputs, or constants from the microprogram unit; thence to the shift unit, and finally to the *d* bus. From the *d* bus, it may be sent to other places, such as the output distributor, microprogram register, etc., or to an arithmetic register.

Data and addressing between memory and the arithmetic unit have their own private channels, leaving the bus free during memory operation. The memory buffer and address register are a part of the arithmetic unit.

Fig. 3 is an expanded view of this unit. Capital letters stand for registers, small letters for logical entities. Registers *A*, *B*, *C* and *E* are simply storage registers, and are used as the Accumulator, *B*-line, Counter and Extension (least significant arithmetic) register. The

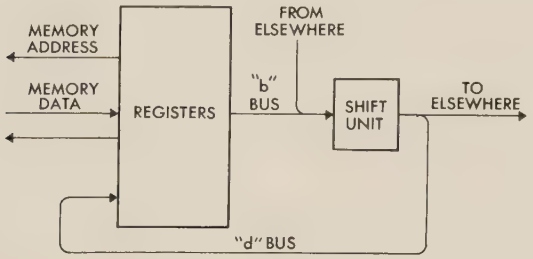


Fig. 2—Arithmetic flow.

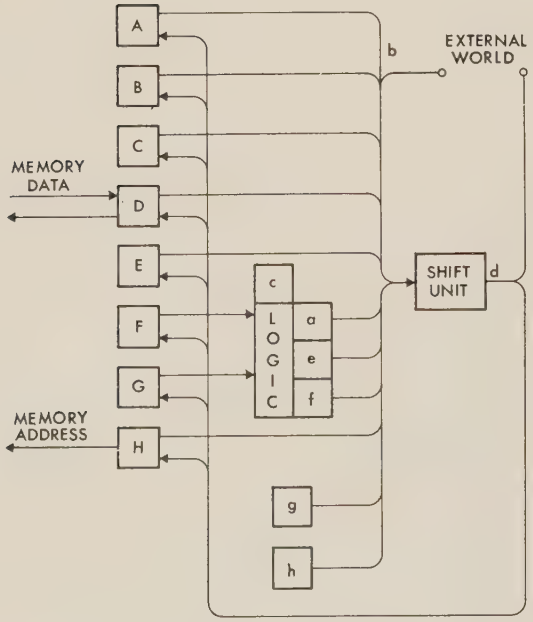


Fig. 3—Arithmetic unit detail.

Distributor, *D*, is the memory buffer, and is often used as working storage. Registers *F* and *G* are the inputs to the adder logic. The *a* logic is the algebraic sum of $(F)+(G)$; *e* is a rather weird logic, ($e=\overline{F}+G$, which is used in generating the extract order); *f*, which yields $F\overline{G}+\overline{F}G$, is used for the "exclusive" or generation; *c* is the carry logic; *g* is a constant emitter, under microprogram control; and *h* is a set of gates used for input.

As a number moves from *b* to *d*, one of five operations may be performed; *viz.*, normal, shift left one bit, shift right one bit, complement or shift left 5 bits. The last is used for automatic fill and in connection with the microprogram unit control.

As an example, to add the number in the *A* and *B* registers, three microprogram steps would be needed. First, transfer *A* to *G*, *D* to *F*, and finally *a* to *A*; 1.5 μsec would be required.

Fig. 4 is a diagram of the microprogram unit. The eight-bit *J* register, augmented by the *TO* flip-flop of the alteration unit, is decoded for up to 512 steps. Students of microprogramming will recognize the Wilkes model in its pure form.[1] The "next" value of the microprogram register may be chosen in one of three ways.

First, the value may be controlled by the microprogram itself.

Second, five bits of the bus, corresponding to the order portion of the word, may be entered; the other three bits are set to zero. In this manner, the order decoding is accomplished.

Third, all eight bits of the *J* register may be filled from the *d* bus. In practice, the order is shifted five bits to the left, presenting eight bits of the address to get the *J* register. In this manner, one may generate "no address" commands.

In principle, the programmer may start on any microstep which amuses him; in practice, only a limited number of these will yield no-address orders, the other steps being used for parts of add, subtract, order procure, etc. The author has no doubt, however, that someone will find a useful reason for popping into the middle of divide or some other command. There is no feature of a machine, however pathological, which cannot be exploited by a programmer.

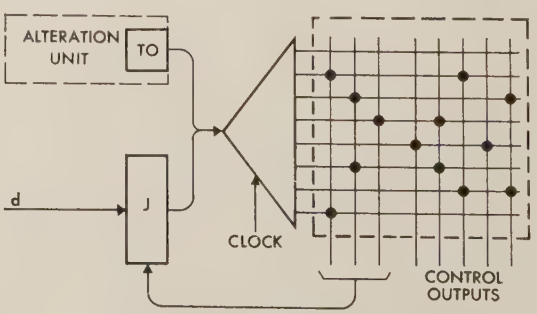


Fig. 4—Microprogram unit.

[1] M. V. Wilkes and J. B. Stringer, "Micro-programming and the design of the control circuits in an electronic digital computer," *Proc. Cambridge Phil. Soc.*, vol. 49, pp. 230–238; June, 1953.

The actual decoding of these nine bits is accomplished
rtly by logic, and partly by current switching of the
ck pulse. A diode matrix is used to convert the micro-
ps into control signals.

No more than 15 micro operations may be called out
a single step, including selection of the next micro
ler.

When stepping the microregister, a ploy is used to
luce the number of diodes. Instead of specifying the
xt step, the microcoder specifies the bits of J which
wishes to reverse. Instead of the minimum latency
ling of earlier days, the microcoder of the SD-2 must
minimum diode coding. This is roughly analogous to
king for a fast, efficient computer program containing
minimum of 1's. The author, as well as others, has
ent endless hours trying to devise a computer pro-
m to do such microcoding, with no results.

One may note in passing that the man who wrote the
crocode, Tomo Hayata, has for several years special-
d in advanced programming problems. Wilkes'
ews,[1] that logical design will in the future be done by
ogrammers, seem to be verified here. Because of the
nited microarithmetic available here, microcoding of
e highest order is a must, since each microstep is 4
ec of time.

For simple orders (*e.g.*, extract), the processes of order
ocure, indexing (but not indirect addressing), operand
ocure and execution can be compressed into the time
two memory cycles, *i.e.*, 40 μsec. Each indirect refer-
ce adds another memory cycle to this time. Only on
ultiply, divide, and shift does the ultra-simple struc-
re begin to be expensive in time.

If the temperature requirement were not imposed, the
ck frequency could be doubled, materially improving
e performance of the machine on multicycle orders.

Fig. 5 is a block diagram of the alteration unit. It
nsists of gates which permit entry of conditions within
e computer or the outside world, flip-flops used as
rking storage, flip-flops, including *TO*, to make its
nclusions known to all and sundry, a five-bit tally
gister (I), a circuit to detect a zero on the d bus, and
e trap logic. There are as many as 20 input gates, 9
rage flip-flops and 10 output flip-flops, exclusive of
).

The I register can change its contents in one of two
ys, *viz.*, counting down by one, or by accepting an
try from the d bus. It may transmit intelligence in
o ways, *viz.*, to the b bus, or by notifying the input
te system that, should anyone care, it has just
unted past zero.

The zero detector signals the truth of the statement
at d is identically zero. In practice, it checks only the
wer digits, not the sign. This is related to the existence
the number -1 in a two-complement system, which
the system's answer to the negative zero of a one's
mplement logic.

The trap logic is as follows: one of the output signals
the alteration unit signals whether or not the system
receiving trap signals; if it is not, the trap logic makes
a note of callers. When the system is again accepting
those signals, it transmits whether or not signals have
been received, and resets its memory to zero. The timing
is such that no trap signal will ever be lost.

The lines going into the logic unit are actually two
busses. Any logic source may read to either bus. The
logic unit has four control wires from the microprogram
unit, specifying which of the 16 Boolean functions of the
two busses is to be put on the output bus. This value is
then routed to the appropriate logic destination.

The output flip-flops have inputs from the logic unit,
and their outputs go to various control points in the
machine. Three major points are: 1) establishing whether
a memory cycle is read/restore or erase/write; 2) setting
the initial carry in the adder; and 3) determining what
value shall shift into the vacant spot on a left or right
shift.

The initial carry is used for more than simply adding
one to a value; since the logic is two complement, but
the one complement one is transmitted on the bus, the
initial carry is, in general, one during subtraction and
zero during addition.

Microprogram Details

Fig. 6 gives circuit details of the microprogram decode
system. The nine flip-flops used are broken into two
groups, one of four, the other of five flip-flops. These
are decoded into, respectively, 16 and 32 wires. In each

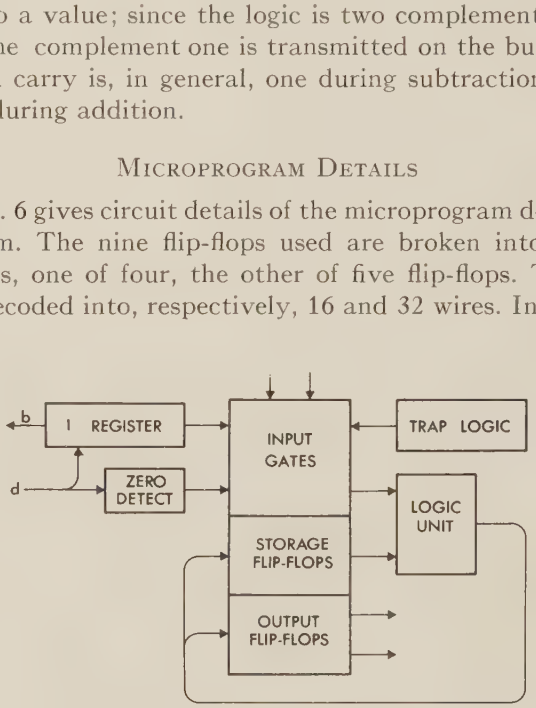


Fig. 5—Alteration unit.

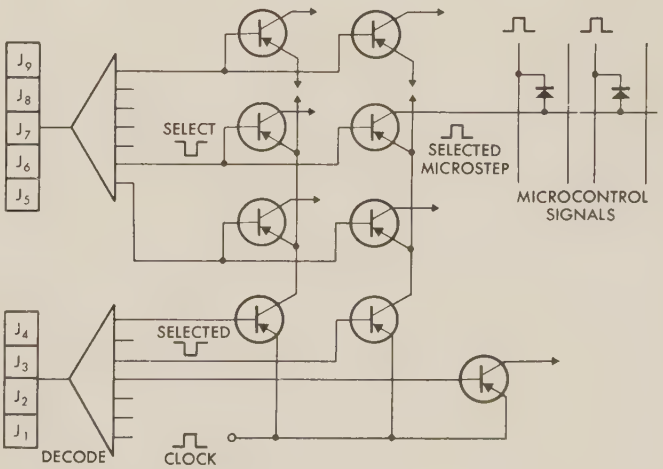


Fig. 6—Details of the microdecode system.

group, one and only one wire goes negative. When the clock signal, of 2 μsec width, is applied to the emitters of the first set of 16 gates, it is passed by the selected gating transistor. From the collector of this transistor, it is routed to the emitter of a set of 32 transistors; again, only one can pass current. Thus, the clock signal is routed to one of $16 \times 32 \times 512$ lines. Diodes on the selected line then cause this signal to be routed to appropriate gates in the arithmetic or alteration unit.

By appropriate placement of diodes, a microstep can operate a variety of gates, the number of which is limited by the current available.

Some of the microcontrol wires return to the *J* register so that the microcoder may control the selection of the next microstep. This register is so designed that the actual change of state is inhibited until the clock goes negative.

While each output of the decoding trees may go to 16 bases, only one transistor of the 16 will have a signal on the emitter; thus only one must be driven.

From an engineering point of view, the control of a computer is an elaborate timing system. A microprogram unit is thus a programmable timing generator. The gating transistor/diode decoding system is but one of many ways to achieve this.

Wilkes has observed[2] that, with the diode system, one has an acute packaging problem. He and his co-workers have been led to consider the use of switch-core decoding.[3]

Eachus[4] and his co-workers have evolved yet another switch-core system which does not depend on coincident current switching.

Order Code

Since the order code is only a small problem in the design of a microprogrammed machine (GOTT SEI DANKE), there is little need to dwell on it. There are several comments of design interest, however.

We were unable, with this structure, to get the multiplication below five microsteps per iteration, nor the divide below six, thus costing respectively 20 and 24 μsec per bit dealt with. Moreover, division required some precalculations (overflow detect) and some postcalculation (obtaining a rounded quotient with a correct remainder) which further boosted its time.

Because of the asynchronous nature of transfer, it is not possible to read into and out of a register simultaneously. Hence, shifting one register requires two steps, or 8 μsec per bit, and double-length shifting requires 16 μsec. This is painful.

Because of the short words, four double-length orders were microprogrammed: add, subtract, clear and add, and store. These take a total of 60 μsec to execute.

A rich collection of branch orders was included. BRanch Unconditionally, BRanch Negative, and BRanch Zero are self-explanatory. BRanch on B is the tally loop order which decreases (B) by one, and branches if it does not go negative. BR1, BR2, BR3, and BR4 are sense toggle branch; if the toggle is set, it is turned off and the program branches. These sense toggles are actually storage flip-flops T1, T2, T3, and T4 of the alteration unit. These may be set by other orders. T1 is also used as an overflow mark.

The machine has a "dynamic" idle. When it is halted either externally or by order, this fact is observed by the microprogram, through the alteration unit, whereupon the microprogram goes into a tight loop, continuously asking, "Can I go? Can I go? Can I go? . . . " Two forms of halting are provided. In "Halt and Display," registers are presented; in the other halt, the console lights are left unaltered. A manual halt is equivalent to halt and display.

For an addressed order, bit positions one through five are sent into the microprogram unit. During order procure, the microprogram examines bits zero and six for indirect addressing and index modification.

A nonaddress order is recognized by the binary equivalent of 31 in the order bits; the microprogram unit causes the order word to shift left 5 bits, and the 8 high bits of the "address" field enter the *J* register.

Conclusion

This paper is not intended to be an argument in favor of the general acceptance of the SD-2 structure as an ideal. Like all computers, the SD-2 is a state-of-the-art device, intended not only to meet the needs of the problems at hand, but also, more importantly, to meet the side conditions of its use. In a vague analogy, the computer specification is like a partial differential equation. The logical designer must choose the boundary conditions and solve the problem, or at least approximate the solution.

With today's emphasis on system speed performance some serious mental gear-shifting on the designer's part is required in order to design a simple machine. It goes against the grain of instinct and experience. *A posteriori* the SD-2 could have been made even simpler, particularly with respect to several peripheral areas not discussed in the paper.

Several conclusions can be drawn here, however. The bus structure is easy to fabricate and maintain; this has been proven on the MILSMAC, a breadboard for the SD-2. It is a highly flexible structure, permitting wide variation in order code with no change in arithmetic unit. At the same time, the components are cascaded to a point where one has the absurd situation of fast switching in a relatively slow computer. A designer of a bus-structured machine would do well to consider alternatives, such as multiple busses, accumulators, etc., to permit more parallelism when speed is important.

[2] M. V. Wilkes, private communication; August 17, 1959.
[3] M. V. Wilkes, W. Renwick, and D. J. Wheeler, "The design of the control unit of an electronic digital computer," *Proc. IEE*, vol. 105, pt. B, pp. 121–128; March, 1958.
[4] Dr. Joseph Eachus of Minneapolis-Honeywell, private conversation; September, 1959.

The use of a special-purpose logic unit, such as the eration unit of the SD-2, gives a freedom of design possible with a special-purpose logic. At the same e, it uses more parts, is slow in handling multiple iable problems, and requires a great deal of control ut. It appears to be a weapon of opportunity.

The use of microprogramming is much the same as general logic unit. Its flexibility and speed of design unquestionable. Also, it uses more parts than a cial-purpose control.

There is no real substitute for a special-purpose de-n. The use of generalized elements in computer design be justified only by the side conditions, never by the basic specification. Where simplicity and speed of design are major items, their use seems indicated.

Wilkes once presented a paper on the best way to design a computer and launched the microprogramming notions. The author would like to comment that if ease and reliability of design are criteria, he was absolutely correct.

Acknowledgment

The author wishes to thank his co-designers, R. Compton and T. Hayata, for their assistance during the design of the SD-2 computer and for their suggestions on this paper.

An Evaluation of Several Two-Summand Binary Adders*

J. SKLANSKY†, SENIOR MEMBER, IRE

Summary—Five fairly representative members of the class of -summand binary adders are described and evaluated. Hopefully, will help the development of more general approaches to com-er subsystems evaluation. The adders are evaluated on the basis hree quantities: the number of two-input AND gates and OR es, G; the gate-normalized addition time, τ; and the number of , n, in each summand. Three plausible formulas for computational iency, η, are postulated, and plotted vs n for the five adders. ed on a comparison of the resulting curves, the following efficiency nula seems preferable:

$$\eta = \frac{n}{\tau \log_2 G}.$$

the five adders considered, the new "conditional-sum adder" is t by the above formula when $n \geq 3$. Other adders, however, are wn to be superior when the assumptions underlying the evalua-of G and τ are changed. The evaluation is found to have several tations; these are discussed. Curves of G and τ vs n are given. s suggested that these curves can serve as raw data for other luations, so that various evaluation methods may be compared.

I. Introduction

AT THE present state of the computer art, adders are essential not only for addition, but also for subtraction, multiplication, and division. As a ult, the speed of any of the so-called "scientific" com-ters depends heavily upon the speed of the adders and associated control equipment. Addition logic is thus of obvious importance, and has received quite a bit of attention from computer designers.

In the present paper, the logic and basic operating principles of several selected adders are discussed, and an evaluation of their performance is made. Several of the problems involved in the construction and application of methods for the present evaluation are also discussed. It is hoped that this paper will lay the groundwork for more general approaches to subsystem evaluation.

II. How Two-Summand Addition Fits into the General Class of Addition Processes

The addition process seems to operate at at least four levels of computer organization:

1) the two-summand level,
2) the multiple-summand level,
3) the operation level,
4) the problem level.

At the two-summand level, the designer views addition as just the addition of two summands, either sequentially or in parallel. At the multiple-summand level, the designer takes advantage of certain logical properties peculiar to the addition of several summands at a time. (An example of a design at the multiple-summand level is separate carry storage [1], [2].) At the operation level, an increase in computational speed is attained through the simultaneous execution of arithmetic and

* Received by the PGEC, December 2, 1959; revised manuscript eived, March 31, 1960.
† RCA Labs., Princeton, N. J.

nonarithmetic operations. At the problem level, microprogramming enables the user to adapt the accuracy or speed of the addition mechanism to the type of problem or data-processing request being fed to the computer.

We see, then, that the two-summand level is the lowest or most microscopic level of addition. As a result, the hardware associated with two-summand adders is most easily isolated from the rest of the computer. This isolation contributes to the ease with which the adders may be evaluated and compared. Unfortunately, even two-summand adders are tied to the rest of the machine through many channels. Consequently, quite an impedance to a satisfactory evaluation exists even at this level. Some of the specific difficulties involved in the evaluation will be discussed in the latter portion of this paper.

III. How Binary Addition Fits into the General Class of Addition Processes

Most adders are binary in two senses: 1) the system of number representation is the binary number system, and 2) the signals processed and transmitted in the adder are two-valued.

Clearly, adders can (and do) exist which are not binary in either or both of these senses. For example, an adder might make use of multiple-state or even analog devices, and process multivalued signals [5]. Furthermore, the system of number representation, while involving binary digits, may be nonpositional. Examples of such number systems are: 1) a system in which the number of 1's in a string of 1's and 0's equals the value of the number represented, and 2) the residue number system [3], [4].

In this paper we restrict ourselves to adders that are binary in both senses.

IV. The Adders Under Consideration

We shall discuss briefly the basic principles and logic circuitry of the following five types of adders:

1) simple series adder (SSA),
2) simple iterated adder (SIA),
3) independent-dependent carry adder (IDA),
4) distant-carry adder (DCA),
5) conditional-sum adder (CSA).

These adders seem to be representative of the class of two-summand binary adders, where "binary" has both of the connotations discussed in the preceding section. The last of these adders is based upon a new scheme, reported in a companion paper [6].

A. The Simple Series Adder (SSA)

The simple series adder (SSA) is the simplest type of adder. It consists of a "full" or "three-input, two-output" adder whose carry output is fed back to the carry input terminal through a unit delay. This is a direct implementation of the standard recursive equations for binary addition,

$$c_{i+1} = x_i y_i \vee y_i c_i \vee c_i x_i \tag{1}$$

$$s_i = x_i \oplus y_i \oplus c_i$$

$$\equiv c_i(x_i y_i \vee \bar{x}_i \bar{y}_i) \vee \bar{c}_i(x_i \bar{y}_i \vee \bar{x}_i y_i). \tag{2}$$

(Our algebraic symbols, as well as the graphical symbols used in our logic diagrams, are defined in Fig. 1.)

The logic diagram of a well-known type of series adder is shown in Fig. 2. The "full adder" portion of this circuit can be partitioned into two half adders [7] as indicated in Fig. 3.

The advantage of the SSA is its simplicity and its economical use of hardware. This, however, is obtained at the cost of a one-carry-at-a-time computation. The carry-to-carry delay is the principal limitation on the addition speed. To circumvent this limitation, several "parallel addition" or "fast carry" schemes have been devised. The remaining four adders illustrate these schemes.

B. The Simple Iterated Adder (SIA)

The SIA consists of a cascade of full adders. A circuit described by Richards [7] is shown in Fig. 4. Here the carry acceleration is obtained in two ways: a) The carry-to-carry delay is only that of a half adder (assuming FA is realized by Fig. 3), thereby doubling the addition speed over that of Fig. 2.[1] b) There is no need to synchronize or to wait for transients to die down before sending a carry to the next full adder.

The circuit shown has provision for setting c_0 to either 0 or 1. (The latter setting would be used for subtraction by 1's complements.) This provision is inherently available in the SSA. Consequently, all the circuits to be considered will have a similar provision, so that the comparison of the adders may be on a uniform basis.

C. The Independent-Dependent Carry Adder (IDA)

In this adder, extra speed is achieved in two ways: a) the simultaneous initiation of carries at all columns where the carries are independent of preceding carries and b) the use of extra fast carry transmission gates to propagate carries that are unchanged (the "dependent" carries).

The basic concept is explained in the following example:

		1	0	0	1	1	0	1	0	0	0	1
		1	1	1	0	1	0	0	1	1	0	0
Step 1:	1				1	0				0	0	
Step 2:		1	1	1			0	0	0		0	
Step 3:	1	1	0	0	0	0	0	1	1	1	0	1

Summands; Independent Carries; Dependent Carries; Sum.

[1] The SSA of Fig. 2 can also be designed to have a half-adder carry-to-carry delay by phasing the summand and carry bits in such a way that the entrance of x_i, y_i into the adder precedes that of c_i by a full clock cycle. In this paper, however, it is assumed that the SSA has a two-half-adder carry-to-carry delay.

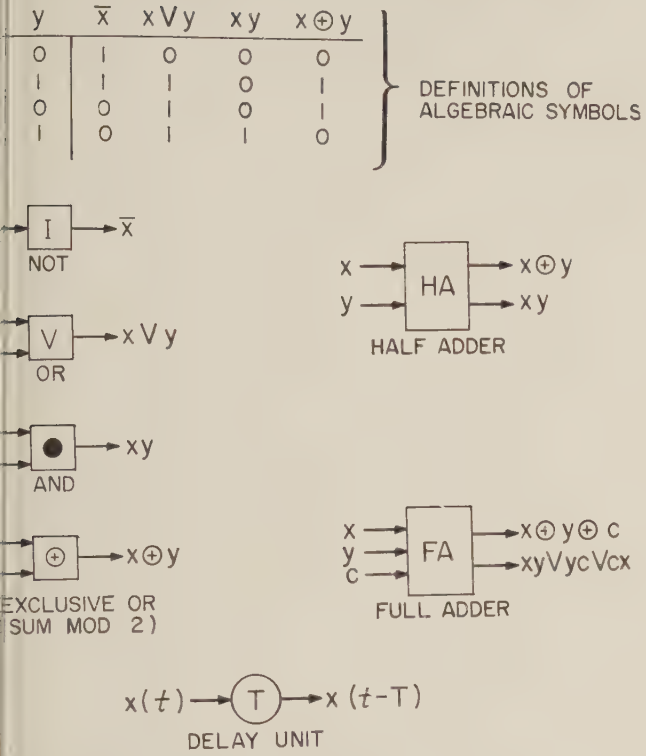
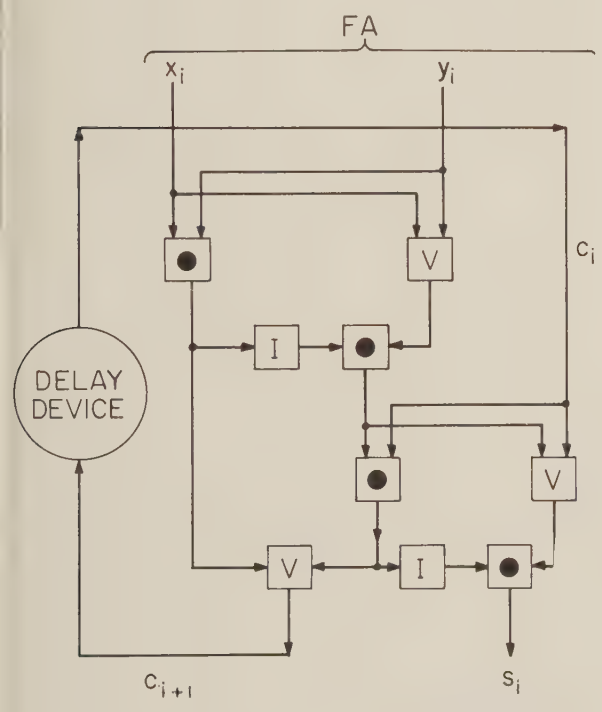


Fig. 1—Algebraic and graphical symbols.



g. 2—An AND-OR-NOT circuit of a simple series adder (SSA).

n Step 1, the existence of independent carries for any rticular column is determined by detecting equality the pair of summand bits of the preceding column; h of these independent carries is then set equal to corresponding summand bits. In Step 2, the remain- carries ("dependent" carries) are generated by ting all the carries between each pair of independent

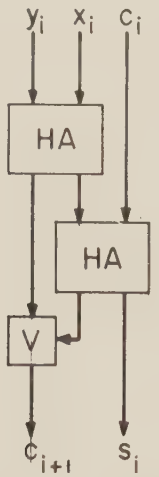


Fig. 3—A full adder decomposed into two half adders and an OR gate.

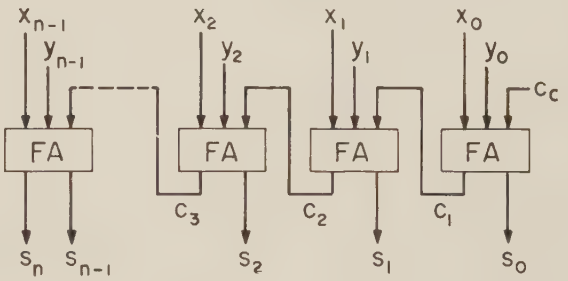


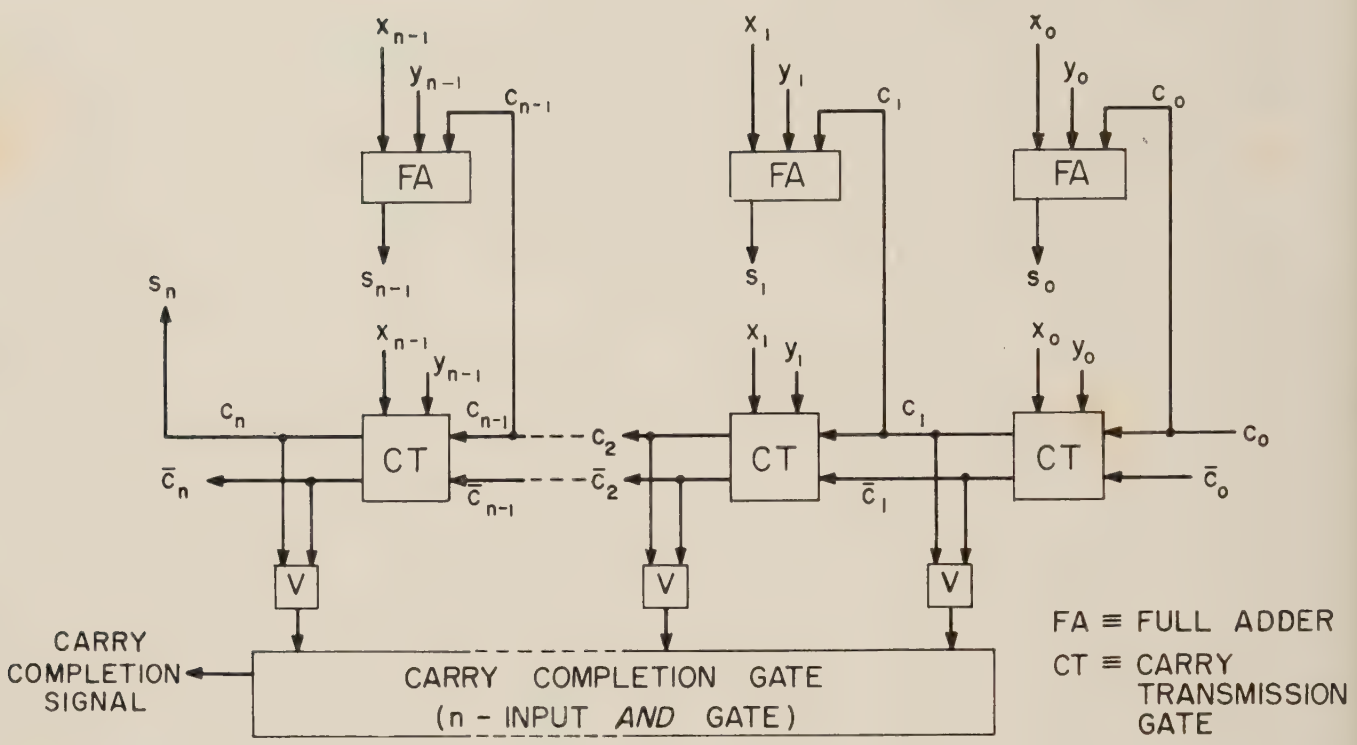
Fig. 4—The simple iterative adder (SIA). "FA" stands for "full adder," for which an AND-OR-NOT realization is given in Fig. 2.

carries equal to the right member of the pair. In Step 3, the carry and the summand bits in each column are added module-2. This yields the sum bit for each column.

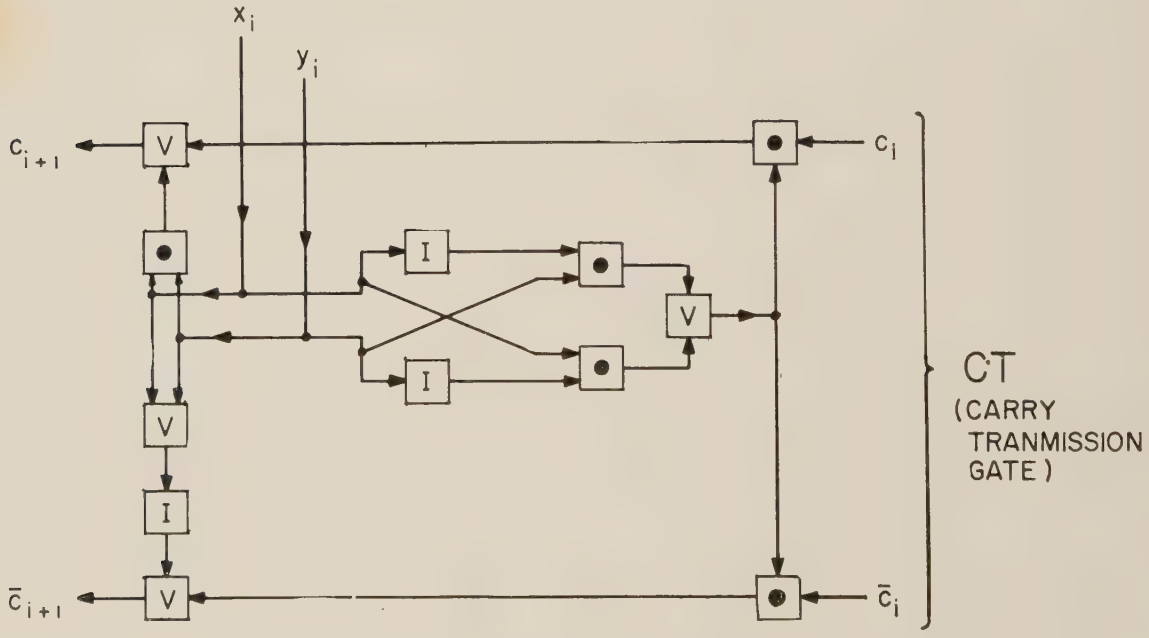
Fig. 5 shows a logic circuit realizing the basic IDA. (This particular circuit was described by Gilchrist, Pomerene, and Wong [8].) The IDA is necessarily an asynchronous circuit, since its addition time depends upon the longest sequence of dependent carries generated by any particular pair of summands. Thus, the circuit in the figure has a carry completion gate, whose purpose it is to transmit to the control circuitry associated with the adder a signal indicating that all the carries have been generated.

D. Distant-Carry Adder (DCA)

Weinberger and Smith [9] have demonstrated that one can construct a fast-carry adder based on a formula expressing the logical dependence of a carry c_i on the pth preceding carry, c_{i-p}, and on the summand bits between columns $i-1$ and $i-p$, inclusive. This formula may be obtained by a series of substitutions into the recursive carry formula, (1). The result is the "triangular" formula,



(a)



(b)

Fig. 5—The indepedent-dependent carry adder (IDA). (a) The over-all circuit; (b) the AND-OR-NOT realization of the carry transmission gate, CT.

$$
\begin{aligned}
c_i = \quad & d_{i-1} \\
& \vee r_{i-1}d_{i-2} \\
& \vee r_{i-1}r_{i-2}d_{i-3} \\
& \cdots\cdots\cdots \\
& \vee r_{i-1}r_{i-2}\cdots r_{i-p+1}d_{i-p} \\
& \vee r_{i-1}r_{i-2}\cdots r_{i-p+1}r_{i-p}c_{i-p},
\end{aligned} \tag{3}
$$

ere

$$
\left.
\begin{aligned}
d_k &\triangleq x_k y_k \\
r_k &\triangleq x_k \vee y_k.
\end{aligned}
\right\} \tag{4}[2]
$$

shall refer to p as the *carry span.*

Weinberger and Smith have shown that the right mber of (3) may be partitioned into a nest of expres-ns, each of which contains variables ("auxiliary ries") which are triangular-formed functions [as in] of other carries or auxiliary carries, and each level auxiliary carry production can have a different carry n. In this way, the carry generation may be further elerated.

The DCA logic becomes clearly advantageous when available switching components consist of multiple-ut AND gates and OR gates, each of which has a ay equal to that of any other gate *regardless of the mber of input terminals on the gate.* In this case, the ry span p is determined primarily by the maximum mber of admissible input terminals per gate.

Figs. 6, 7, and 8 (pp. 218–220 illustrate a class of CA's in which the carry spans at all the levels of aux-ry carries and actual carries are equal. In this paper restrict ourselves to this class of DCA's.

It is deducible from these figures that the addition ne per column declines rapidly as the summand length, increases. Another property of the DCA, which can advantageous, is that its operation is completely nchronous.

Conditional-Sum Adder (CSA)

The logic of the CSA is reported in a companion per [6]. Involved in this logic is the computation of e "conditional" sums and carries that result from the sumption of all possible distributions of carries for rious groups of columns. The operation is completely nchronous, as in the DCA.

It will be shown that the CSA is the fastest of the ders considered, under the constraint that only two-ut AND gates and two-input OR gates are admissi-, and that these gates all have the same individual ays.

The process is illustrated in Fig. 1 of Sklansky [6] for e case of a sixteen-bit addition. Fig. 2 of that paper ows the logic circuit of a seven-bit CSA having pro-ion for setting c_0 to either 0 or 1.

V. Series-Parallel Combinations

Series adders are obviously efficient in their utilization of hardware, but relatively slow. Parallel adders, on the other hand, utilize a substantial amount of hardware in order to achieve a relatively high speed. From the designer's point of view, both the series and parallel types may represent extremes to be avoided. Compromises between the two types are possible in the form of series-parallel combinations.

The following are two approaches to achieving such compromises:

1) The pair of summands is divided into equal segments, each segment q bits in length. Each segment of the summand-pair is sent to a q-bit parallel adder. The adder computes q sum bits and a carry. The carry and the next q-bit summand segment are sent to the adder, and the process is repeated.

2) The carry-to-carry delay may be reduced by the use of distant-carry logic (Figs. 9 and 10). The summand bits arrive at the adder one column-pair at a time, each column-pair following the preceding pair by d time units. Suppose the available AND gates and OR gates each impose a delay greater than d. In that case, the carry fed back to the input will necessarily be removed from the incoming summand pair by one or more columns. The distant-carry logic shown in Fig. 10 generates the carry of the column succeeding that of the input carry, and the time required to generate this carry is less than or equal to d. (In this figure, $d=1$.) In this way, the rate of processing one-column-at-a-time data can be increased beyond that permitted by a simple series adder. (This approach was suggested by Fowler [10].)

VI. Evaluation Criteria

As mentioned earlier there are many difficulties in constructing good methods of evaluating adders, mainly because an adder is tied to the rest of a computer through many channels, each channel imposing its own set of constraints.

We approach evaluation-method construction through the concepts of *investment*, I, and *return*, R.[3] In terms of these concepts we define the adder's *efficiency*, η, as

$$\eta = \frac{R}{I}. \tag{5}$$

Our object, of course, is to maximize the efficiency.

We have isolated three variables that we feel should enter the definitions of R and I:

1) The number of two-input AND-gates and two-input OR-gates, G.
2) The gate-normalized computation time, τ; *i.e.*, the number of gates involved in the longest chain of gates through which a sequence of signals travels. (In the case of the IDA, which is asynchronous,

[2] $\triangleq$ means equals by definition.

[3] These terms were suggested by S. Amarel of RCA Laboratories.

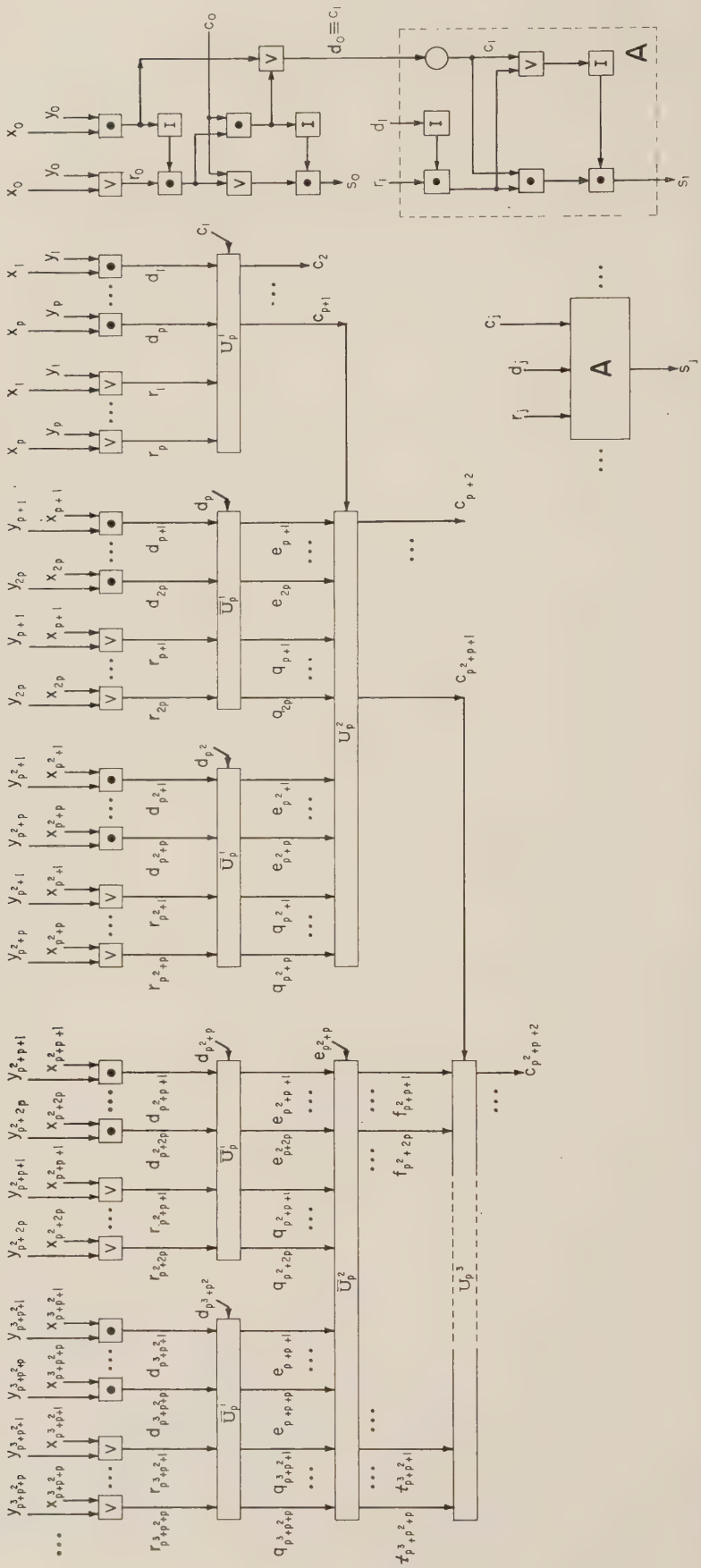


Fig. 6—The logic circuit of a distant-carry adder (DCA) in which the carries and the auxiliary carries at all levels are produced with carry spans equal to p. The AND-OR-NOT circuit for A is shown inside the dashed rectangle at the right. The AND-OR-NOT circuits for U_4^1, $\bar{U}_4^1$, and $\bar{U}_4^2$ are indicated by examples in Figs. 7 and 8. From these examples the general circuit, namely the circuit for U_p^i, may be inferred.

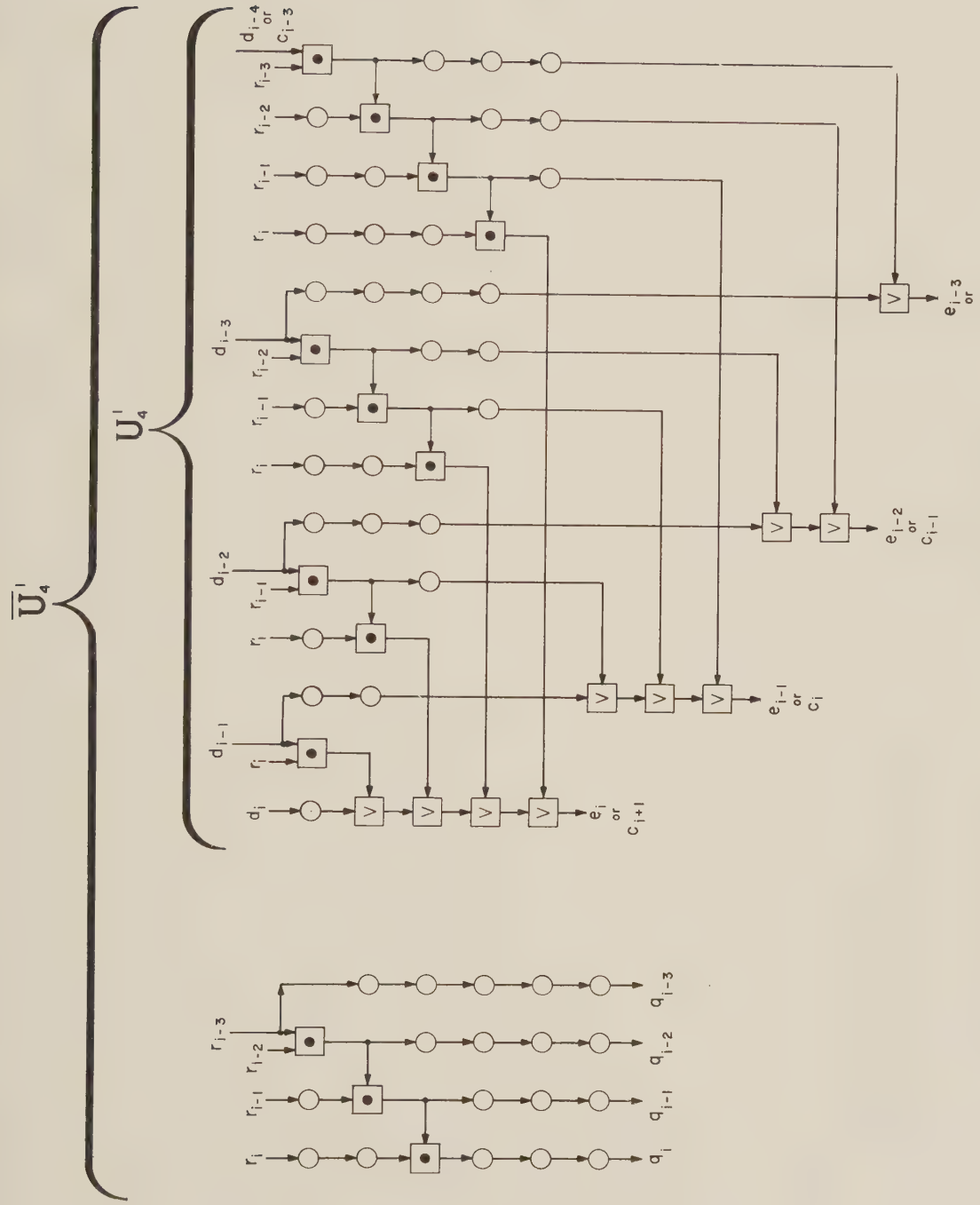


Fig. 7—The AND-OR-NOT circuit of $\bar{U}_4^1$. A portion of this circuit realizes U_4^1; this is indicated by the braces. ($\bar{U}_4^1$ and U_4^1 appear in the logic circuit of a distant-carry adder; see Fig. 6.) The circular blocks represent unit delays.

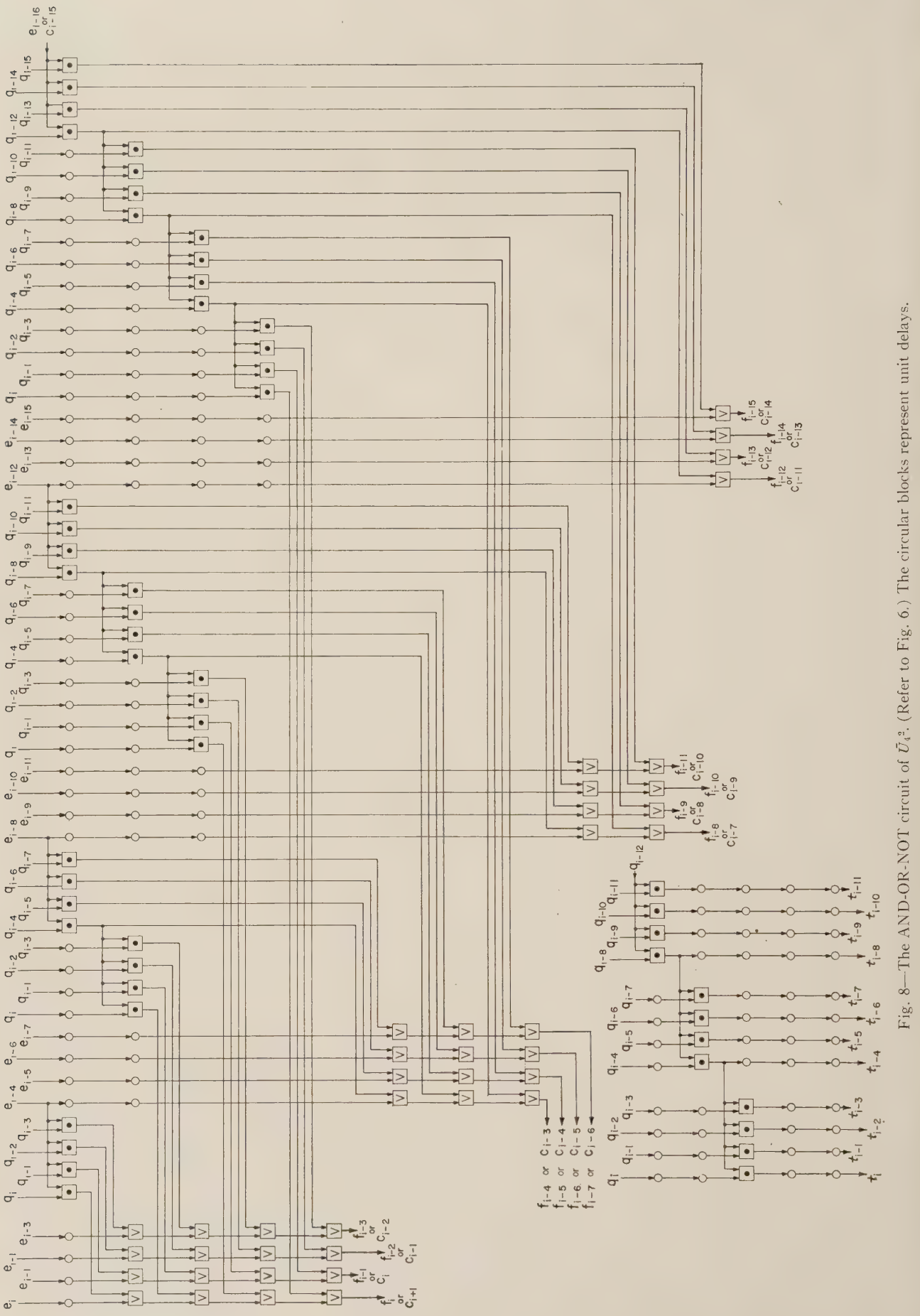
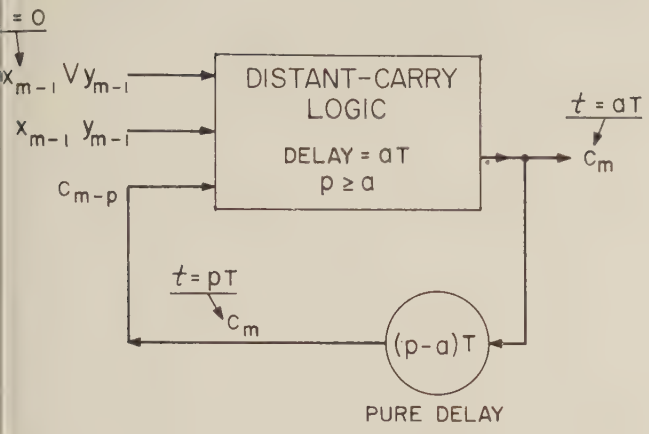
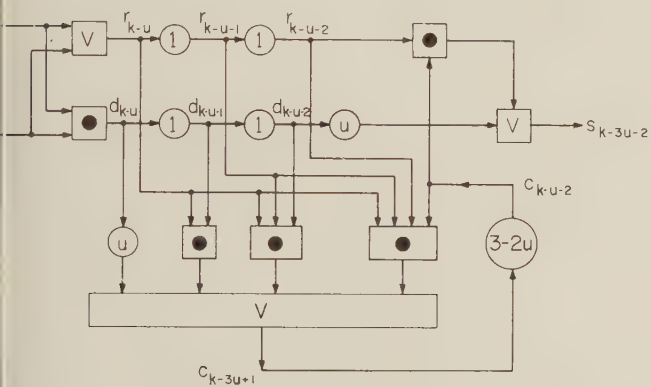


Fig. 8—The AND-OR-NOT circuit of $\bar{U}_4^2$. (Refer to Fig. 6.) The circular blocks represent unit delays.



. 9—Fast carry-to-carry sequences implemented by distant-carry logic. Here the carry delay is a times the interval between the arrival of successive pairs of summand bits. Each variable in this figure is labeled with a value of time, t, referred to the reference time, $t=0$.



. 10—The AND-OR-NOT circuit of a series adder based on the scheme of Fig. 9 for the case of $p=3$. (Necessary assumption: $p \geq 2u$.) The letter u represents the delay of each AND gate and OR gate. The variables x, y, r, d, c, and s are defined in Section IV.

we computed bounds on the average value of τ. The derivation is in the Appendix.)

3) The maximum number of summand bits, n, that the adder can accept.

Curves of G and τ vs n have been computed for the e adders, and plotted in Figs. 11 and 12. The formulas these curves are given in Table I. (Two values of the rry span, $p=2$ and $p=5$, were assumed for the DCA.) ese formulas are based upon the logic circuits given Figs. 2 to 8 of this paper, and Fig. 2 of Sklansky [6].[4] Certain constraints were assumed in the definitions G and τ. These are:

1) The logic circuits of the adders consist of only o-input AND gates, two-input OR gates, and NOT tes. No time-domain devices such as flip-flops, emory cells, delay units, pulse generators, etc., are mitted. (As a result of the two-input-gate constraint, ery multiple-input AND gate or OR gate was replaced

[4] In the case of the IDA, the expression $4+n$ given for τ in the le, and upon which the curves are based, is actually a lower bound. known upper bound on the *average value* of τ is $4+n+2\log_2 n$, ich is, percentagewise, not much larger than $4+n$ (see Appendix).

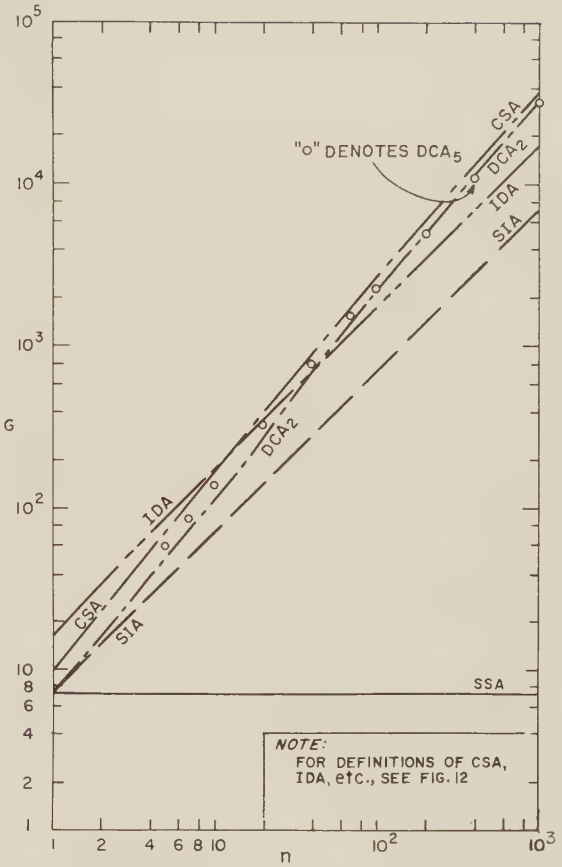


Fig. 11—Number of two-input AND gates and OR gates, plotted as functions of n.

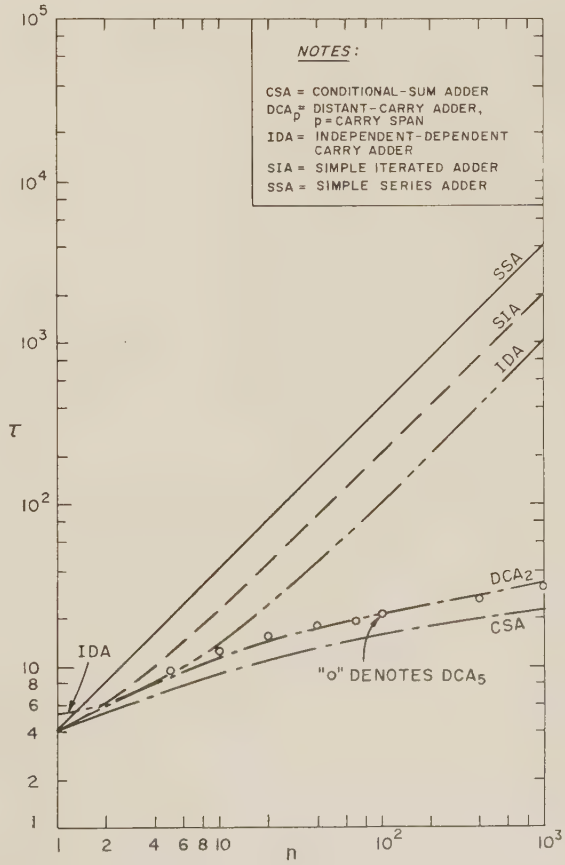


Fig. 12—Gate-normalized addition time, τ, plotted vs n.

TABLE I

THE FORMULAS FOR G AND τ FOR THE FIVE ADDERS IN TERMS OF THE SUMMAND LENGTH, n

Type of Adder	Number of Gates G	Number of Gate Delays τ
Simple Series Adder (SSA)	7	$4n$
Simple Iterated Adder (SIA)	$7n$	$2(n+1)$
Independent-Dependent Carry Adder (IDA)	$17n-1$	$4+n$ (lower bound)
Distant-Carry Adder (DCA_p) $p \triangleq$ carry span	$6n+1+\frac{p+1}{p-1}q+\frac{p^2+2p-1}{p}\left[k\left(n+\frac{1}{p-1}\right)-\frac{pq}{(p-1)^2}\right]$, where $k \triangleq \log_p[1+n(p-1)]-1$, $q \triangleq 1+(n-1)p-n$	$4+k(p+1)$, where $k \triangleq \log_p[1+n(p-1)]-1$
Conditional-Sum Adder (CSA)	$3n[2+\log_2(n+1)]$	$2+2\log_2(n+1)$

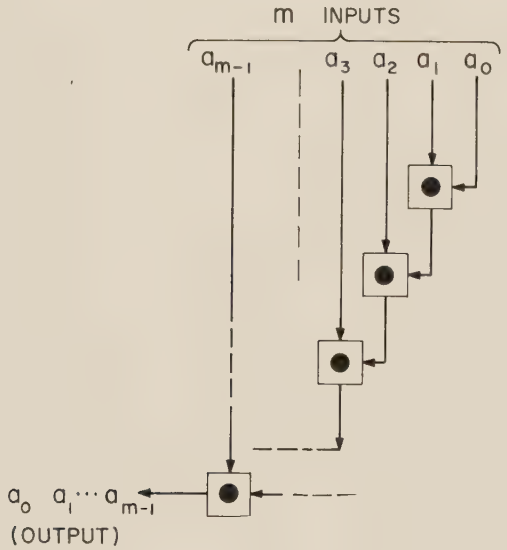


Fig. 13—A cascaded two-input AND gate realization of an m-input AND gate.

by a network of cascaded two-input gates, such as that shown in Fig. 13.)

2) The NOT gates are not counted in evaluating G, and they all have zero delay.

In the present investigation we arbitrarily define $R=n$, and consider the following three possibilities for I: τ, $\tau \log_2 G$, and τG. This results in the following three criteria:

$$\eta = \frac{n}{\tau} \tag{6}$$

$$\eta = \frac{n}{\tau \log_2 G} \tag{7}$$

$$\eta = \frac{n}{\tau G}. \tag{8}$$

These three formulas were evaluated for the five adders with $p=2$ and $p=5$ chosen for the DCA. The results are plotted in Figs. 14, 15, and 16.

VII. Choosing Among the Criteria

An examination of Figs. 14 to 16 will show that the relative standings of the adders are sensitive to the criteria. Hence, a need exists for choosing among the criteria. The following three qualitative observations lead to a preference for (7). We hope more scientific approaches to criterion selection will some day be available.

1) (6) is unacceptable because it gives no weight at all to the investment of hardware. It is not conceivable that unlimited amounts of hardware can be free, from the point of view of maintenance as well as manufacture.

2) (8) is ruled out because it selects the simple series adder, SSA, as "best." The computer market is full of evidence that computer designers are willing to pay an increase in hardware for a less than proportionate increase in addition speed.

3) Mass production and mass maintenance techniques tend to reduce the unit cost of hardware when the hardware contains groups of large numbers of identical units. This is another factor indicating that (8) should be ruled out.

As a result of these observations, (7) seems preferable to the other two for general-adder-design purposes. We leave open the possibility that one or both of the other two criteria may be significant under special conditions.

VIII. Some Comparative Observations

Under (7), the conditional-sum adder (CSA) is seen to be more efficient than any other adder when $n \geq 3$. By the curves of (6), the CSA is shown to be faster than all the adders when $n>1$, and at least as fast as any

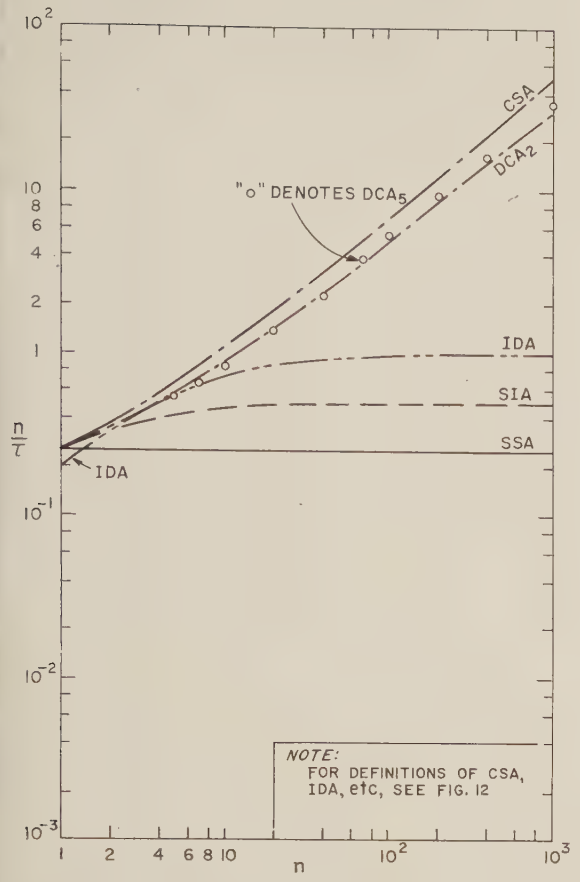


Fig. 14—Comparison of the adders on the basis of $\eta = n/\tau$—(6).

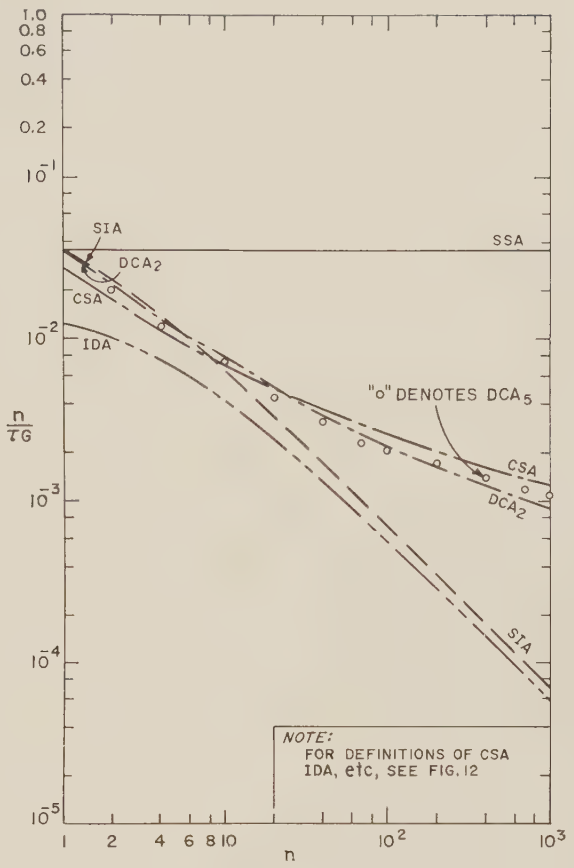


Fig. 16—Comparison of the adders on the basis of $\eta = n/(\tau G)$—(8).

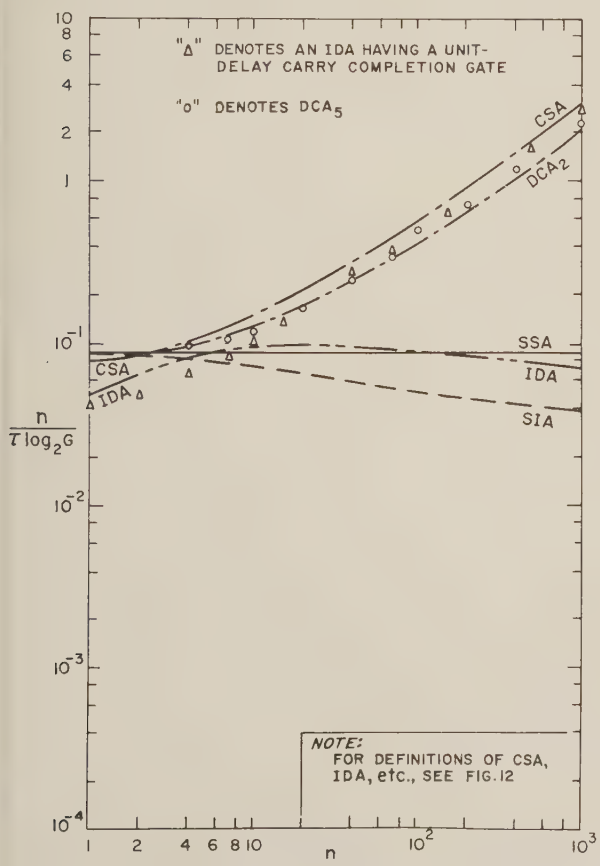


Fig. 15—Comparison of the adders on the basis of $\eta = n/(\tau \log_2 G)$—(7).

other adder when $n = 1$. Under (8), the SSA is "best."

Fig. 11 shows that the curves of G for all the parallel adders fall within the band $7n < G < 28n$ when $n < 100$. Thus, an approximation for G within a factor of 2 is

$$G \cong 14n \quad \text{when} \quad n < 100. \tag{9}$$

The DCA and CSA are close on all the curves. A significant advantage of the CSA over the DCA is in speed: the CSA is about 50 per cent faster. The DCA, on the other hand, requires a slightly smaller number of gates.

We note that the curves of the DCA are relatively insensitive to the carry span. This is due to the replacement of multiple-input gates by cascades of two-input gates.

IX. Limitations of the Present Evaluation

An evaluation based directly or indirectly on the G-τ-n relationship must be qualified by a substantial number of factors. These factors fall in two categories: *device considerations*, and *systems-matching considerations*.

Under device considerations we note:

1) In practice, multiple-input AND gates and OR gates are often available. If we admit such gates in our analysis, the relative standings of the adders may change. This comment has particular significance for the IDA and DCA.

To illustrate, suppose the carry completion gate of the IDA is realizable by a multiple-input AND gate having a normalized delay of 1. In that case, an approximation of the average value of the IDA's gate-normalized addition time would be

$$\tau = 4 + 2a_n \cong 6 + 2 \log_2 n, \tag{10}$$

rather than that given by Table I or inequality (13) of the Appendix. (See Appendix for the definition of a_n and for the derivation of an approximate formula for a_n.)

Suppose further that we assign to this carry completion gate a value of G equal to that of an equivalent cascaded AND gate network. (This seems reasonable, since the cost of a multiple-input AND gate can be expected to increase with the number of input terminals.) As a result, the curve of η for (7) would appear along the triangles shown in Fig. 15. Thus, a fast carry completion gate has brought the IDA's efficiency from a relatively low level (near the efficiencies of the SSA and SIA) to a relatively high level (near those of the CSA and DCA), especially for large n. Clearly our assumptions about the implementation of the carry completion gate are quite important (Fig. 17).[5]

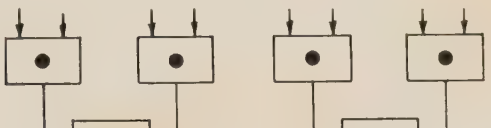


Fig. 17—A multiple-input gate in which the number of two-input-gate delays is $\log_2 n$.

2) The IDA's speed might be increased by implementing the carry transmission gates (defined in Fig. 5) by relay-like devices. In devices of this type, the inertia to be overcome when shifting the state of a transmission gate from a stopping to a transmitting state would be substantially greater than that required to transmit the carry through an "open" gate, *i.e.*, a gate in a transmitting state. (We note, however, that this situation is probably not realizable in ultrafast circuits where space and time are strongly correlated.)

The advantage in speed gained by this implementation will be especially strong when the carry completion gate is realized by a fast device, as in 1) above. Thus, if the dependent carries are generated k times faster than the independent carries, (10) becomes

$$\tau = 4 + \frac{2}{k} a_n = 4 + \frac{2}{k} + \frac{2}{k} \log_2 n. \tag{11}$$

[5] An intermediate curve for the IDA will be obtained if the carry completion gate is assumed to be implemented in the form shown in Fig. 17. In this gate the normalized delay is $\log_2 n$.

3) Many devices, such as "tunnel" diodes and parametric phase-locked oscillators, are able to realize *threshold logic* efficiently. These devices would require completely new adder circuits, together with new expressions for G and τ and a new set of curves of η.

Under systems-matching considerations we note:

1) An adder should be matched to the arithmetic unit. Thus, special circuits such as those required for overflow, end-around carry, various complementation schemes, and the tie-in with subtraction, multiplication, and division, must all be considered.

2) An adder should be matched to the memory organization. The series adder, for example, goes better with a variable-word-length memory than do any of the parallel adders. In the case of fixed-word-length memories, the present analysis would encourage the use of long summand lengths (since, when (6) and (7) are applied to the DCA and the CSA, the resulting curves of η increase with n), while practical considerations involving the instruction format would encourage shorter word lengths.

3) An adder should be matched to the control circuitry. A significant aspect of the control circuitry is the organization of timing. Two classes of timing are usually distinguished, "synchronous" and "asynchronous," corresponding to central and local timing control, respectively. For asynchronous control the IDA is particularly appropriate, while the other adders are suitable for synchronous control.

The remaining systems-matching considerations given below relate to systems within which the entire computer is imbedded. Even though these systems occupy higher hierarchic levels than that of the adder, still their requirements do reflect on the final choice of adder design.

4) An adder should be easy to design. An adder associated with a relatively simple design technique will require less expensive engineering than one requiring difficult design methods. This consideration reflects on the cost of the adder, although the exact amount may be difficult to estimate.

5) An adder should be easy to fabricate. An adder requiring a large number of different components is likely to be more expensive to manufacture than one built with only one or two basic building blocks. An adder requiring many complicated connections between components may also be difficult to manufacture.

6) An adder should have good reliability, and be easy to maintain. This consideration will be affected strongly by the particular maintenance system adopted. If, for example, the entire adder is replaced periodically on a preventive-maintenance schedule, the reliability requirements on the adder will probably not be as severe as those for an adder in which basic logic gates are replaced on a corrective maintenance basis.

X. General Observations and Conclusions

The concept of efficiency has a certain theoretica

raction. This attraction consists of the hope of some y finding a general formula for information-processing ciency with which it will be possible to obtain a aximum realizable" efficiency for any specified task information processing [12, 13]. This paper demonates, however, that any such "maximum" will be a ximum only within a special region bounded by merous practical qualifications.

Using an evaluation of adders as an example, we have ustrated some of these qualifications and the resultant fficulties involved in assigning a meaningful efficiency a subsystem of a computer. Many restrictive assumpns were found necessary in order to have reasonably nple efficiency formulas. Especially restrictive were ose limiting the admissible components to two-input ND gates and OR gates, and those ignoring the asciated control circuitry. In spite of these limitations, wever, it was possible to obtain a few general comrative observations that will be meaningful under rtain conditions. An example of such conditions are ra-high-speed computers, where time and space are rongly correlated.

Within the stated assumptions, the conditional-sum der is superior to the other adders whenever $n \geq 3$ ig. 15). The two DCA's follow closely behind the CSA, wever, and may be suitable for some situations, ecially where multiple-input gates are available at sacrifice in delay time per gate. The CSA is also own to be faster than the other adders when $n > 1$, d no slower than any of them when $n = 1$ (Fig. 14). The independent-dependent carry adder will have rong advantages if extra fast carry completion signals d/or extra fast "dependent" carry transmissions can realized. This, however, will probably not be possible ultrafast circuits having strong space-time corre-ions.

More flexible approaches to adder evaluation are pos-le when it is desired to have a number of free vari-les in the efficiency formulas. Under these conditions, may be convenient to use the concept of *loss*, taken m economics, and express loss as a sum of *costs*. We ggest that these other approaches can make use of e curves of G and τ given in this paper. One may then mpare various evaluation methods, and perhaps help e evolution of evaluation methodology.

Appendix

Derivation of Bounds on the Gate-Normalized Addition Time of the IDA

The IDA circuit upon which this derivation is based shown in Fig. 5. This circuit includes a carry comple-on gate, which is necessary in order to detect the rmination of the carry sequences. These carry se-uences are of variable length, depending upon the par-cular summands.

The carry completion gate is a multiple-input AND te. Hence, in order to satisfy the constraints imposed our definitions of G and τ (Section VI), this gate is replaced by a cascade of two-input AND gates of the form shown in Fig. 13. With this replacement, the formula for $G(n)$, namely

$$G(n) = 17n - 1, \tag{12}$$

is easily found.

On the other hand, a formula for $\tau(n)$ is not so easily obtained. We were able to find only lower and upper bounds on τ. These are $4+n$ and $4+n+2 \log_2 n$, *i.e.*,

$$4 + n < \tau < 4 + n + 2 \log_2 n. \tag{13}$$

In this expression, τ of necessity represents an average value because of the asynchronous nature of the mechanism.

The lower bound, $4+n$, is a lower bound not only on the average value of τ, but also on any specific instance of τ. This results from the fact that a signal traveling from one end of the carry completion gate to the other consumes $n-1$ gate delays. This time, plus the five gate delays between any pair of summand input terminals on a carry transmission gate and the corresponding terminal of the carry completion gate results in $4+n$ as the lower bound.

The derivation of the upper bound depends on a result obtained by Burks, Goldstine, and Von Neumann. In their now classical report [11], they demonstrated that the average longest sequence of carries of value 1 in the addition of two summands of length n is less than, but not much less than, $\log_2 n$. $\text{Log}_2 n$ thus serves as an approximation for the average length of longest 1-carry sequences.

Unfortunately we cannot apply this result directly, because the circuit we have chosen for analysis propagates 0-carries as well as 1-carries. (This dual carry circuitry is needed for the implementation of asynchronous operation [8].) For this circuit the average longest carry sequence, a_n, may be estimated by noting that a_n is on the average equal to the longest 1-carry sequence in *pairs* of n-length additions, since the statistics of 0-carries are identical to those of 1-carries. Now the longest sequence among 0- and 1-carries in a pair of n-length summations will usually equal the longest 1-carry sequence in the $2n$-length summation of the double-length summands obtained by placing the two addition matrices side by side. We say "usually" because of the occasional longest carry sequence that covers the boundary between the original two n-length addition matrices. However, these exceptions are relatively rare, especially for large n. The effect of neglecting these exceptions means that our estimate of a_n is made somewhat larger. Hence, we have

$$a_n \leq \log_2 (2n) \equiv 1 + \log_2 n. \tag{14}$$

(For $n = 40$, $1+\log_2 n = 6.3$. The actual value of a_{40} is 5.6). Hereafter we shall treat $1+\log_2 n$ as an approximation, rather than merely upper bound, for a_n.

We note that the average value of τ necessarily is less than

a) the delay incurred in entering the summand bits into the carry transmission gates, plus
b) the average propagation time of the longest carry sequence through the carry transmission gates, plus
c) the delay through an OR gate leading into the carry completion gate, plus
d) the longest possible delay through the carry completion gate.

Delay a) is 2. Delay b) is $2a_n$, caused by the delay of 2 in each carry transmission gate. Delay c) is 1. Delay d) is $n-1$. Hence,

$$\tau < 2 + n + 2a_n. \tag{15}$$

Substituting (14) into (15) yields

$$\tau < 4 + n + 2\log_2 n, \tag{16}$$

which completes our derivation.

Acknowledgment

The author is indebted to Dr. S. Amarel of RCA Labs. for many discussions on the general approaches to subsystems evaluation.

Bibliography

[1] A. H. Taub, *et al.*, "On the Design of a Very High-Speed Computer," Digital Computer Lab., University of Illinois, Urbana, Rept. No. 80; October, 1957.
[2] G. Metze, "A Study of Parallel One's Complement Arithmetic Units with Separation Carry or Borrow Storage," Digital Computer Lab., University of Illinois, Urbana, Rept. No. 81, November 11, 1957.
[3] H. L. Garner, "The residue number system," IRE Trans. on Electronic Computers, vol. EC-8, No. 2, pp. 140–147; June 1959.
[4] A. Svoboda, "Rational numerical system of residual classes," *Stoje Na Zpracování Informaci*, vol. 5, pp. 9–37; 1957.
[5] J. Sklansky, "Analog and threshold building blocks for variable-radix adders and other logic networks," to be published in *Trans. AIEE (Commun. and Electronics)*.
[6] J. Sklansky, "Conditional-sum addition logic," IRE Trans on Electronic Computers, this issue, pp. 226–231.
[7] R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Princeton, N. J., 1955.
[8] B. Gilchrist, J. H. Pomerene, and S. Y. Wong, "Fast carry logic for digital computers," IRE Trans. on Electronic Computers, vol. EC-4, pp. 133–136; December, 1955.
[9] A. Weinberger and J. L. Smith, "A one-microsecond adder using one-megacycle circuitry," IRE Trans. on Electronic Computers, vol. EC-5, pp. 65–73; June, 1956.
[10] F. H. Fowler, Jr., internal report, RCA.
[11] A. W. Burks, H. H. Goldstine, and J. Von Neumann, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," The Institute for Advanced Study, Princeton, N. J.; June 28, 1946.
[12] D. E. Muller, "Complexity in electronic switching circuits," IRE Trans. on Electronic Computers, vol. EC-5, pp. 15–19; March, 1956.
[13] K. Steinbuch, "Die logische Verknüpfung als Einheit der Nachrichtenverarbeitung," *Nachrtech. Z.*, vol. 12, pp. 169–175; April 1959.

Conditional-Sum Addition Logic*

J. SKLANSKY†, senior member, ire

Summary—Conditional-sum addition is a new mechanism for parallel, high-speed addition of digitally-represented numbers. Its design is based on the computation of "conditional" sums and carries that result from the assumption of all the possible distributions of carries for various groups of columns.

A rapid-sequence mode of operation provides an addition rate that is invariant with the lengths of the summands. Another advantage is the possibility of realizing the adder with "integrated devices" or "modules."

The logic of conditional-sum addition is applicable to all positive radices, as well as to multisummand operation.

In a companion paper, a comparison of several adders shows that, within a set of stated assumptions, conditional-sum addition is superior in certain respects, including processing speed.

* Received by the PGEC, December 2, 1959; revised manuscript received, March 31, 1960.
† RCA Labs., Princeton, N. J.

I. Introduction

CONDITIONAL-sum addition is a new scheme of parallel, high-speed addition for digital computers. A comparative evaluation of several binary adders[1] indicates that the conditional-sum adder (CSA) is quantitatively superior in certain important respects, including computation speed.

In the present paper the basic concepts of CSA logic are presented, and a specific AND-OR-NOT network realizing the CSA is described.

[1] J. Sklansky, "An evaluation of several two-summand binary adders," this issue, p. 213.

II. Normal Mode of Operation

We explain the normal operation of CSA by the following example.

Fig. 1 shows the conditional-sum addition of two binary-coded numbers:

$$x = 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1$$

$$y = 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0.$$

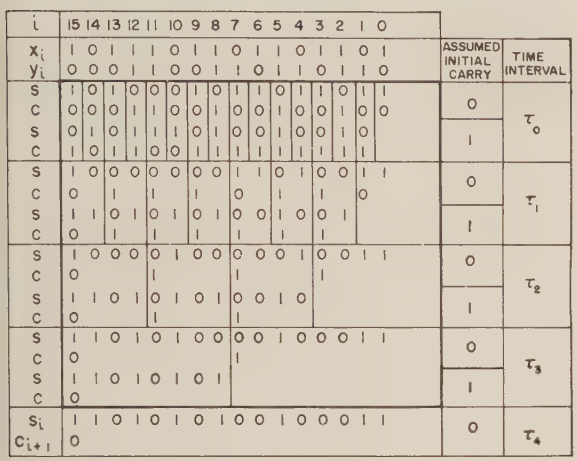


Fig. 1—Example of a conditional-sum addition.

We represent by τ_0, τ_1, τ_2, τ_3, τ_4 the successive time intervals during which "conditional" sums and carries are computed. During τ_0 two sum-and-carry pairs for each column are computed: one pair under the assumption (assumption A_0) that the carry brought to each column is 0, and the other pair under the assumption (assumption B_0) that the carry brought to each column is 1. An exception to this is at column 0, where only A_0 is assumed, since the carry brought to that column is known to be 0. The first row belonging to τ_0 in Fig. 1 contains the conditional sums under assumption A_0, the second row has the conditional carries under the same assumption, and the third and fourth rows contain the conditional sums and carries under assumption B_0. In particular, consider column 0. Here we have, using the well-known Boolean relations for sum and carry bits,

$$s_0^0 = x_0 \oplus y_0 = 0 \oplus 1 = 1$$

$$c_1^0 = x_0 \cdot y_0 = 0 \cdot 1 = 0.$$

(We define our symbols in Appendix I.) For column 1 we have

$$s_1^0 = x_1 \oplus y_1 = 1 \oplus 0 = 1$$

$$c_2^0 = x_1 \cdot y_1 = 1 \cdot 0 = 0$$

$$s_1^1 = s_1^0 \oplus 1 = 0$$

$$c_2^1 = x_1 \vee y_1 = 1 \vee 0 = 1$$

In a similar manner we find the other entries in the τ_0 columns.

During τ_1 conditional sums and carries for *pairs* of columns (namely, column numbers 0 and 1, 2 and 3, $\cdots$, and 14 and 15) are computed simultaneously, under the assumptions (A_1) that the carry brought to each pair of columns is 0, and (B_1) that this carry is 1. The rows belonging to τ_1 are arranged in a manner similar to those of τ_0.

Continuing this process, the "conditional" sums and final carries are computed for tetrads of columns during τ_2, for octads of columns during τ_3, and for the entire sixteen-column group during τ_4. During τ_4 the sums and final carry produced are no longer "conditional," but are equal to the sum bits and the final carry for the two summands, x_i and y_i.

To understand the process in more detail, consider the pair of τ_1 columns (0, 1) (referred to hereafter as a "τ_1 array"). The entries for the upper half of τ_1-column 0 are the same as those of the upper half of τ_0-column 0, since the entries in these halves both correspond to the case $c_0 = 0$. However, since the carry for column 0 will no longer be needed, only the sum bit need be entered in the τ_1 array. The entries in the upper half of τ_1-column 1 are found from τ_0-column 0 in the following way: If the second entry of τ_0-column 0 is 0—*i.e.*, if $c_1^0 = 0$—then the upper half of τ_1-column 1 is identical to the upper half of τ_0-column 1; if $c_1^0 = 1$, then the upper half of τ_1-column 1 is identical to the *lower* half of τ_0-column 0. The reason for this is that the upper half of any τ_0 column corresponds to a zero-valued incoming carry for that column, while the lower half corresponds to a one-valued incoming carry. Thus, since τ_0-column 0, *viz.*,

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

has a second entry of 0, it follows that τ_1-column 1 is identical to the upper half of τ_0-column 1, *viz.*,

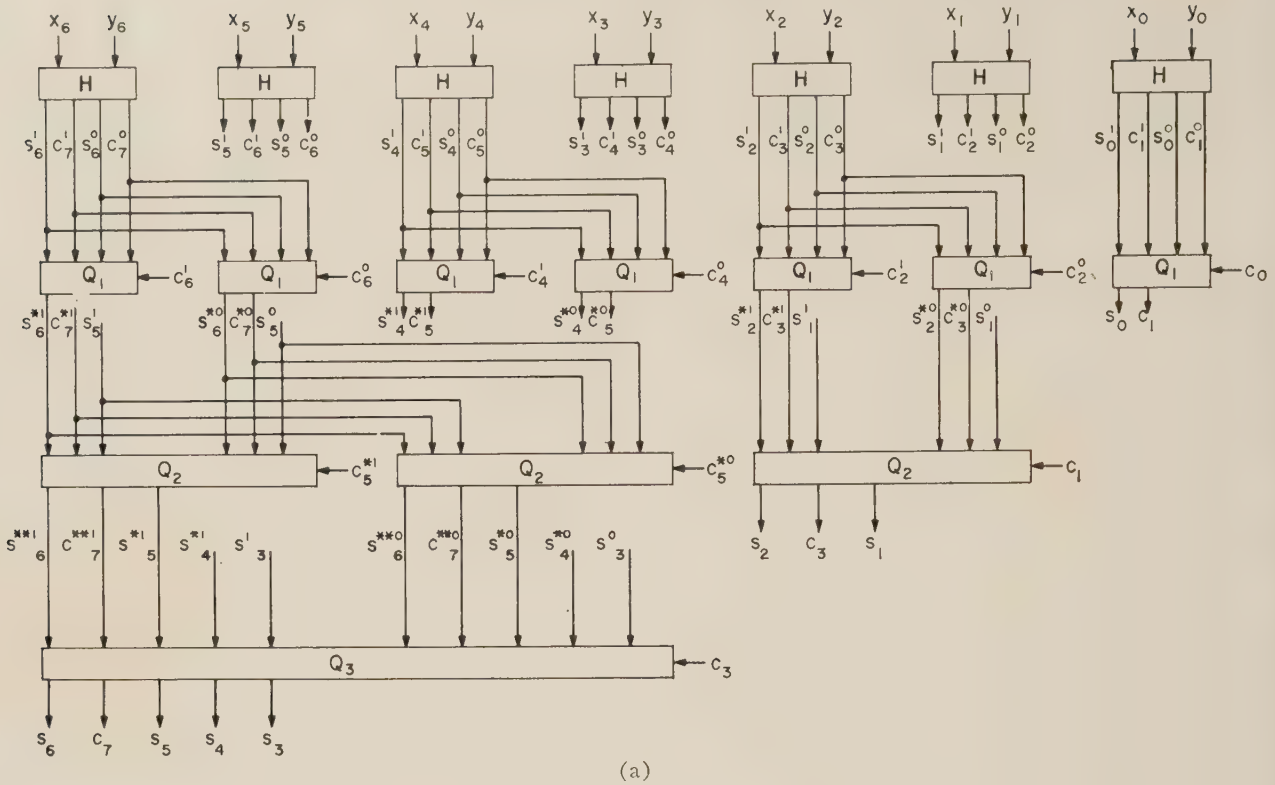
$$\text{upper half of } \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

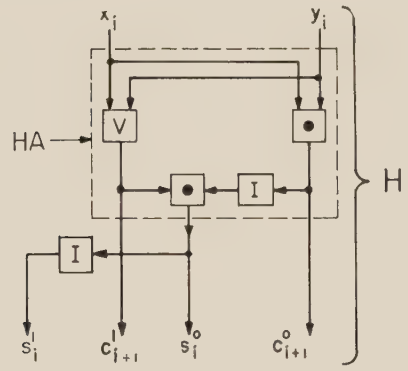
Summarizing,

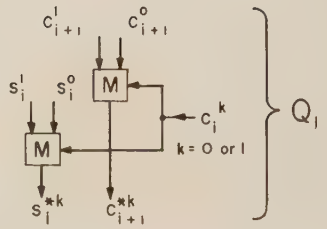
$$\tau_0 \text{ array} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & \\ 1 & \end{bmatrix} \Rightarrow \tau_1 \text{ array} \begin{bmatrix} 1 & 1 \\ 0 & \\ & \\ & \end{bmatrix}.$$

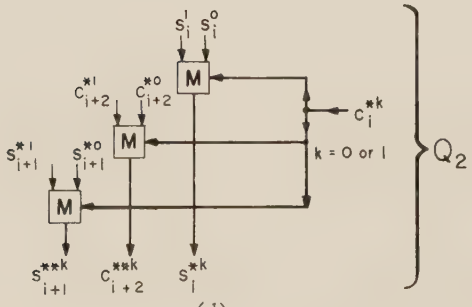
In a similar manner,

$$\tau_0 \text{ array} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \Rightarrow \tau_1 \text{ array} \begin{bmatrix} 0 & 0 \\ 1 & \\ 0 & 1 \\ 1 & \end{bmatrix}.$$









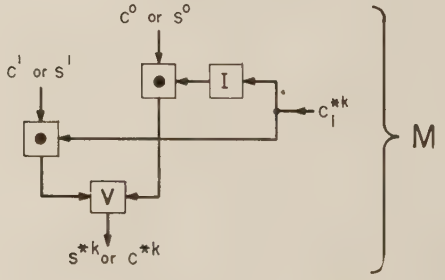


Fig. 2—The logic circuit of a seven-bit conditional-sum adder. (a) The over-all circuit; (b) an AND-OR-NOT circuit for H, a portion of which is the circuit for HA; (c) and (d): Q_1 and Q_2 in terms of M; (e) an AND-OR-NOT realization of M. The circuits for Q_3 and other Q_i's can be inferred from (c) and (d).

In the above situation, the lower half of the τ_1 array is determined by the lower right-hand carry bit in the τ_0 array, *i.e.*, the bit in the lower right-hand corner.

For later intervals, τ_j, the operation is similar except that greater numbers of columns are involved at each step. For instance, to obtain columns 4 to 7 of interval τ_2 the following transformation takes place:

$$\tau_1 \text{ array} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & & 1 & \\ 0 & 0 & 1 & 0 \\ 1 & & 1 & \end{bmatrix} \Rightarrow \tau_2 \text{ array} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & & & \\ 0 & 0 & 1 & 0 \\ 1 & & & \end{bmatrix}.$$

For each interval τ_j the columns in Fig. 1 are marked in groups of 1, 2, $\cdots$, 2^i, $\cdots$ according to the size the group of columns participating in the transformation from τ_{j-1} to τ_j.

III. Rapid-Sequence Mode of Operation

In the event several sums are to be computed successively, one may increase the effective speed of the adder by storing the results of cycle τ_j for use during cycle τ_{j+1}; *simultaneously* another set of conditional sums and carries could be computed for a different pair of summands. Consequently, the addition speed of this "rapid-sequence" mode of operation would be faster than that of the normal mode by the factor

$$D = \frac{1}{\tau_R}(\tau_0 + \tau_1 + \cdots + \tau_j + \cdots + \tau_p). \qquad (1)$$

where $\tau_R \triangleq$ pulse repetition period. When $\tau_R = \tau_0 = \tau_1 = \cdots = \tau_p$, this expression reduces to

$$D = p + 1. \qquad (2)$$

Since $n = 2^p$ (see Fig. 1), D can be expressed in terms of n:

$$D = \log_2 2n. \qquad (3)$$

IV. A Suggested Logic Circuitry

The suggested logic circuitry, indicated in Fig. 2, consists entirely of AND gates, OR gates, NOT gates, and their interconnections. The timing and storage circuitry for the controlling sequence of operations is omitted. Actually this timing and storage circuitry is not necessary for the basic operation of adding two summands. However, when many summands are to be added in rapid sequence, in the manner discussed in the previous section, then it is advantageous to have timing control. Storage circuitry is needed here only for guaranteeing the proper synchronism of signals; if the AND gates and OR gates imposed pure delays with no signal distortion, and if the delays of all the AND gates were exactly the same, then no storage circuitry would be necessary.

Fig. 2(a) shows the information-flow diagram for a seven-column adder; adders for greater numbers of columns can be inferred from the figure.

Q_1 and Q_2, shown as blocks in Fig. 2(a), may be realized in terms of a basic module, M, whose AND-OR-NOT circuit is given in Fig. 2(e). The suggested M realizations for Q_1 and Q_2 are given in Fig. 2(c) and 2(d). The Q_i's for $i>2$ can be realized by networks easily inferred from the realizations of Q_1 and Q_2. A suggested AND-OR-NOT realization of H is given in Fig. 2(b).

For neatness, not all the connections are shown explicitly as continuous lines. For instance, the signals s_1^0 and s_1^1, produced by the H of column 1, are brought to the input terminals of Q_2 of column-pair (1, 2). These connections are indicated in the figure by labels on the input and output leads.

A warning to circuit designers: the maximum "fan-out" (the number of input leads emanating from an output terminal) is an increasing function of the summand length. This can be verified from an examination of Fig. 2, especially parts (c) and (d). The "fan-out" is an index of the load that a gate must be capable of handling.

V. Extensions of the Basic Concepts

The basic concepts of conditional-sum addition can be extended in at least the following two directions: 1) Positional number systems with radices greater than 2, in which a numeral $x_N \cdots x_1 x_0$ represents the number

$$\sum_{i=0}^{N} x_i r^i,$$

where r, a positive integer, is the radix of the number system. 2) Multiple-summand addition, in which more than two summands are added simultaneously.

A. Higher Radices

It is possible to apply conditional-sum addition to higher-radix number systems with little change in the basic concepts. We illustrate this in Fig. 3 by an example in the decimal system. The alphabetical symbols here are the same as in Fig. 1, and the description of the operation is similar to that given in Section II for the binary case.

i	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ASSUMED INITIAL CARRY	TIME INTERVAL
X_i	2	6	7	7	4	1	0	0	2	6	9	2	4	3	5	8		
Y_i	5	6	0	4	9	7	9	4	1	5	1	7	1	6	4	5		
S	7	2	7	1	3	8	9	4	3	1	0	9	5	9	9	3	0	τ_0
C	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0	1		
S	8	3	8	2	4	9	0	5	4	2	1	0	6	0	0		1	
C	0	1	0	1	1	0	1	0	0	1	1	1	0	1	1			
S	8	2	8	1	3	8	9	4	4	1	0	9	5	9	0	3	0	τ_1
C	0		0		1		0		0		1		0		1			
S	8	3	8	2	3	9	9	5	4	2	1	0	6	0			1	
C	0		0		1		0		0		1		0					
S	8	2	8	1	3	8	9	4	4	2	0	9	6	0	0	3	0	τ_2
C	0				1				0				0					
S	8	2	8	2	3	8	9	5	4	2	1	0					1	
C	0				1				0									
S	8	2	8	2	3	8	9	4	4	2	0	9	6	0	0	3	0	τ_3
C	0								0									
S	8	2	8	2	3	8	9	5									1	
C	0																	
S_i	8	2	8	2	3	8	9	4	4	2	0	9	6	0	0	3	0	τ_4
C_{i+1}	0																	

Fig. 3—Example of a conditional-sum addition in the decimal system

B. Multisummand Addition

The scheme of conditional-sum addition may be applied to the simultaneous addition of more than two summands. This may be done by storing and computing a conditional sum and a conditional carry for each possible value of an incoming carry at each appropriate column. For instance, in four-summand addition the possible incoming carries for any column are 0, 1, 2, and 3. Conditional sums and carries must be computed for each of these four carries during each of the cycles τ_j.

For the general case of p-summand addition the possible carries are 0, 1, $\cdots$, $p-1$, no matter what the

radix may be.[2] Thus it should not be difficult to synthesize a conditional-sum adder that will handle both multiple summands as well as radices greater than 2.

VI. High Speed

In a companion paper[1] we show that the CSA is basically faster in processing speed than several other well known adders. The primary assumptions in that analysis are

1) That the fundamental building blocks are a two-input AND gate, a two-input OR gate, and a one-input NOT gate;
2) That the AND gate and OR gate impose equal delays, while the NOT gate imposes no delay.

VII. The Possibility of Using "Integrated Devices"

We note that the conditional-sum logic circuitry just described lends itself to realization by a large number of identical "integrated devices" or "modules" (labeled M in Fig. 2). The use of "modular" construction is likely to be economically attractive, especially with the recent development of "integrated" solid-state devices.[3]

[2] This is proved in Appendix II.
[3] J. T. Wallmark and S. M. Marcus, "Integrated devices using direct-coupled unipolar transistor logic," IRE Trans. on Electronic Computers, vol. EC-8, pp. 98–107; June, 1959.

VIII. Conclusions and Summary

Conditional-sum addition has the following attractive properties:

1) In the "normal" mode of operation, the addition time *per column* is a decreasing function of the number of columns, n. (This time is roughly $(\log_2 2n)/n$ gate delays.)

2) In the "rapid-sequence" mode of operation, the time between the production of successive n-bit sums is invariant with respect to n.

3) Both the normal and the rapid-sequence modes of operation are completely synchronous, so that the control circuitry associated with a CSA adder should be simpler than for comparable asynchronous mechanisms.

4) The logic circuit suggested for CSA contains many identical AND-OR-NOT subnetworks, thereby lending itself to realization by "integrated devices" or "modules."

5) CSA may be extended to other modes of arithmetic, specifically: multisummand addition and non-binary radices.

6) CSA proves superior to other schemes in addition logic in certain quantitative respects, including processing speed.[1]

APPENDIX I

Definitions of Symbols

$\oplus$ = plus, modulo 2.
$\cdot$ = AND.
$\vee$ = INCLUSIVE OR.
i = column number, beginning with column 0.
j = ordinal number of an interval, τ_j, during which an array of conditional sums and carries are produced.
x_i, y_i = summand bits of column i.
c_i = carry bit entering column i.
s_i = sum bit of column i.
c_i^0 = carry generated at column $i-1$, assuming $c_{i-1}=0$.
c_i^1 = carry generated at column $i-1$, assuming $c_{i-1}=1$.
c_i^{*0} = carry generated at column $i-1$, assuming $c_{i-2}=0$.
c_i^{*1} = carry generated at column $i-1$, assuming $c_{i-2}=1$.
c_i^{**0} = carry generated at column $i-1$, assuming $c_{i-3}=0$.
c_i^{**1} = carry generated at column $i-1$, assuming $c_{i-3}=1$.
$c_i^{\overbrace{*\cdots*}^{k}0}$ = carry generated at column $i-1$, assuming $c_{i-k-1}=0$.
$c_i^{\overbrace{*\cdots*}^{k}0}$ = carry generated at column $i-1$, assuming $c_{i-k-1}=1$.
s_i^0 = sum generated at column i, assuming $c_i=0$.
s_i^1 = sum generated at column i, assuming $c_i=1$.
$s_i^{\overbrace{*\cdots*}^{k}0}$ = sum generated at column i, assuming $c_{i-k}=0$.
$s_i^{\overbrace{*\cdots*}^{k}1}$ = sum generated at column i, assuming $c_{i-k}=1$.

The graphical symbols used in the figures are self-explanatory.

Appendix II

A Theorem for Multisummand Addition

In an addition of any p positive summands, the carry oduced by any column has a maximum possible value $p-1$.

This result is independent of the radix, r, and includes e case of a carry consisting of more than one digit, ., a carry $\geq r$. There is no restriction on the length of e summands.

Proof: Consider column 0 . The maximum possible lue of its sum is $pr-p$. We ask the reader to verify at the corresponding output carry of this column is $-p_1$, where p_1 is the positive integer satisfying the ophantine inequality

$$(p_1 - 1)r \leq p \leq p_1 r \tag{4}$$

e., p_1 is 1 plus the number remaining after the lowest-der digit of the r-ary representation of $p-1$ is de-ted). As a consequence of (4) and the fact that $r>1$, follows that

$$1 \leq p_1 \leq p. \tag{5}$$

he maximum possible sum produced by column 1 is erefore $pr-p_1$. Hence, by an analysis similar to that used for column 0, we conclude that the corresponding output carry of column 1 is $p-p_2$, where p_2 is the positive integer satisfying

$$(p_2 - 1)r \leq p_1 \leq p_2 r. \tag{6}$$

Hence,

$$1 \leq p_2 \leq p_1 \tag{7}$$

Continuing in this manner, we find that the maximum possible input carry of column k is $p-p_k$, where

$$(p_k - 1)r + 1 \leq p_{k-1} \leq p_k r \tag{8}$$

and that

$$1 \leq p_k \leq p_{k-1} \leq \cdots \leq p_2 \leq p_1 \leq p. \tag{9}$$

Hence the output carry of column k cannot exceed $p-1$. Since k is arbitrary, the theorem is proved.

Acknowledgment

The author is indebted to L. S. Levy and W. H. Cherry of RCA for ideas leading to the present work.

Constant-Weight Counters and Decoding Trees*

WILLIAM H. KAUTZ†, member, ire

***Summary*—A class of counters is described in which the number 1's in the flip-flops or register stages composing the counter re-ins constant as the counter advances from state to state. Simple gital circuit arrangements are described for the design of such unters, which may be used with a particular type of decoding tree economical ring-type counters, to provide a separate output lead each state. Some interesting theoretical questions concerning the nimization of these decoding trees are raised and partially an-ered. Finally, the costs of these counters are compared with one other, and with those of other types of counters, over a continuous ge of values of the flip-flop/gate-input cost ratio.**

I. Introduction

AN N-digit ring counter may be viewed as a simple circulating shift register of N stages; it contains a binary number having only a single 1, the ner digits being 0. Such a counter counts with period as shift pulses are applied to the register. The N out-ts which indicate the successive states of the counter are derived directly from the stages of the register, as shown in Fig. 1.[1]

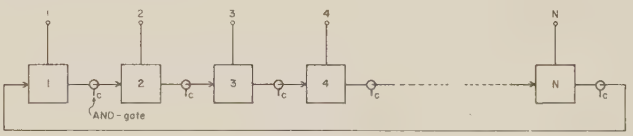


Fig. 1—A simple ring counter.

This paper describes some other ring-type counters based on this shift-register viewpoint. These counters are similar to ring counters in that an individual output is provided for each of the counts, but they differ from ring counters in that more than just a single 1 is circulated. The separation of the two or more 1's is varied on successive circulations by means of auxiliary logical connections between the stages of the register. As a re-

* Received by the PGEC, December 7, 1959.
† Stanford Research Institute, Menlo Park, Calif.

[1] The interstage AND gates shown in this and following figures are intended to be symbolic only, and represent any of several known techniques for shifting binary digits from one register stage to another in response to a counting pulse c.

sult, the number n of stages (binary storage elements) is generally much less than the total count N. A network of AND gates is required to provide the N outputs (one per count) from the n stages.

It will be shown that for a given count N, an over-all economy of equipment is often possible through the use of this type of counter. Moreover, in many applications not all N outputs of an N-state ring-type counter are actually needed, so that some of the output AND gates can be eliminated in these cases if a shift-register type of realization is used. Further, it is not uncommon in ring-counter applications for the outputs to drive AND-gate loads, in the capacity of these counters as timers for a sequence of digital events. With these types of counters, the output logic might therefore be accomplished merely with one or more additional inputs on these AND-gate loads, instead of with separate AND gates.

In the following two sections, circuit arrangements are described for register-type counters that circulate *two* and *three* 1's, respectively, as the counter advances from state to state. The network of AND gates required to decode the register stages into individual state outputs is then derived, with the objectives of minimality in the number of gate inputs, and a simple and systematic network structure. In Section V, an application of these counters to the reduction of the number of drive lines in multiple-coincidence memories is described. Constant-weight counters are then compared with one another and with more conventional counters to determine the conditions under which each is preferable. The appendixes contain some of the more intricate derivations and proofs, including the proof that p/n counting codes cannot be digit-weighted.

II. 2-out-of-n Counters

For purposes of illustration, consider a 2-out-of-5 counter, that is, a 5-digit shift register that is to be modified in some way to count through all 10 states whose binary numbers have exactly two 1's. A simple circulating register can count to only 5 with such a restraint, but can do it in two ways:

01100	01010
00110	00101
00011	10010
10001	01001
11000	10100
(01100)	(01010).

The inter-1 spacing is 0 and 1 for the two cases (or 3 and 2, respectively, depending on which of the two 1's is taken to be the first). One way to count to 10 would be to modify the counter somehow to pass from the last state of each of these two sequences to the first state of the other; that is, to force the transitions:

$$11000 \rightarrow 01010$$

$$10100 \rightarrow 01100.$$

For each of these transitions, the left-hand 1 shifts in the normal way, but the right-hand 1 either shifts an extra stage, or does not shift at all. One condition for this special transition to occur is that the first stage contain a 1. This condition may be seen to be sufficient by noting that for the two other states in the sequences which contain 1's in the first stage (namely, 10001 and 10010), those stages which would be modified to effect the special transitions (stages 2 and 3) contain 0's only, and will therefore remain unchanged. Thus, to count to 10, the register should circulate while the first stage contains a 0, and shift according to the special transitions whenever the first stage contains a 1.

If we designate the binary variables of the contents of the register as $x_1, x_2, \cdots, x_5$, left to right, and the succeeding (next) values of these variables as $x_1', x_2', \cdots, x_5'$, then for a pure circulating register, $(n=N=5)$, the sequential circuit equations are[2]

$$x_1' = x_5$$
$$x_2' = x_1$$
$$x_3' = x_2$$
$$x_4' = x_3$$
$$x_5' = x_4.$$

For the modified register $(n=5,\ N=10)$:

$$x_1' = x_5$$
$$x_2' = x_1$$
$$x_3' = \bar{x}_1 x_2 + x_1 x_3$$
$$x_4' = \bar{x}_1 x_3 + x_1 x_2$$
$$x_5' = x_4$$

($\bar{x}_1$ is the complement of x_1; + designates inclusive OR, and juxtaposition designates AND). Thus, when $x_1=0$, these equations reduce to the first set above, and the register behaves as a pure circulating register. When $x_1=1$, stages 1, 2, 4, and 5 behave as a circulating register, but stage 3 does not shift (*i.e.*, it stays as it is) and stage 2 shifts around stage 3 to stage 4.

The complete counter is shown in Fig. 2. (The network of output gates at the top is described in Section IV.) The condition $x_1=0$ may be seen as a connection from the 0 side of the first stage to the interstage transfer gates, 2→3 and 3→4. The condition $x_1=1$ reveals itself only as a connection from the 1 side of the first stage to the interstage transfer gate 2→4. The absence of all advance clocks associated with stage 3 when $x_1=1$ prevents this stage from shifting.

The technique just described for $n=5$ may be applied to any number n of stages to achieve a count N equal to the total number of n-digit binary numbers that have exactly two 1's; namely,

$$N = \binom{n}{2} = \frac{n(n-1)}{2}.$$

[2] W. H. Kautz, "Stage-logic relations in autonomous sequential networks," *Proc. EJCC*, pp. 119–127; December, 1958.

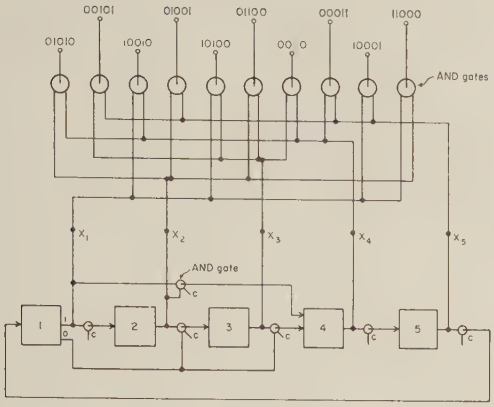


Fig. 2—A 2/5 ring-type counter complete with decoding tree.

Starting with the state having 1's in stages 2 and k is given below), the counting is accomplished by de-easing by one the inter-1 spacing after each n counts. his is done by inhibiting the shift of the right-hand 1 henever the left-hand 1 is in the first stage of the gister. Eventually, the inter-1 spacing will become ro. Finally, when state 1100 . . . 0 is reached, the ter-1 spacing is increased in the next shift, thereby oving the two 1's to stages 2 and k again. When n is d, the count is made up of exactly $(n-1)/2$ sequences states, each sequence consisting of n states with a ed inter-1 spacing

$$\left(0,\ 1,\ 2,\ \cdots \frac{n-3}{2}\right).$$

hen n is *even*, there are $(n-2)/2$ such sequences con-ting of n states each, and one more short sequence th $n/2$ states and an inter-1 spacing of $(n-2)/2$, *e.g.*, e sequence starting with 010010 for $n=6$, or with 000100 for $n=8$.

The general equations are:

$$\begin{aligned}
x_1' &= x_n \\
x_2' &= x_1 \\
x_3' &= \bar{x}_1 x_2 + x_1 x_3 \\
x_4' &= \bar{x}_1 x_3 + x_1 x_4 \\
&\vdots \\
x_{k-1}' &= \bar{x}_1 x_{k-2} + x_1 x_{k-1} \\
x_k' &= \bar{x}_1 x_{k-1} + x_1 x_2 \\
x_{k+1}' &= x_k \\
&\vdots \\
x_n' &= x_{n-1}.
\end{aligned}$$

ere

$$k = \left[\frac{n+4}{2}\right],$$

here the brackets indicate the integral part of the uantity within them. The register, exclusive of the utput gates, is shown in Fig. 3.

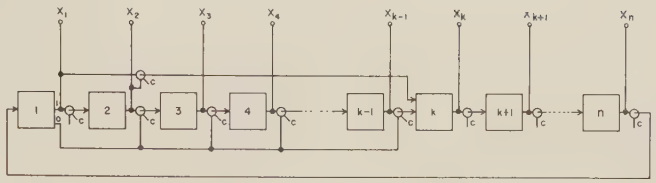


Fig. 3—General form of a $2/n$ ring counter.

Other equations are possible, all based on the principle of modifying the inter-1 spacing after each n counts; they differ only in the permutation of the last subscripts in the equations for x_3' through x_k'. The set given above is probably the simplest, however, since all of the special transitions except one can be handled by simply inhibiting the advance clock to about half of the stages of the register. The remaining transition requires a single feed-forward path from stage 2 to stage k.

If the counter need not recycle after counting to N, the feed-forward path may be eliminated. The counter is then started in the state having 1's in stages 2 and k, which is the state that would follow the use of the feed-forward path if it were present.

In some practical register circuits, a circuit simplification is possible if one can be assured that the register will never contain two adjacent 1's. This can be achieved for the counter in question if the 3rd and kth equations above are replaced by

$$x_3' = x_2 \quad \text{and}$$

$$x_k' = \bar{x}_1 x_{k-1} + x_1 x_3, \qquad \text{respectively.}$$

Thus, the feed-forward transition follows state 10100 · · · 0 instead of 1100 · · · 0, and the total count is $N=n(n-3)/2$ instead of $n(n-1)/2$.

It should be noted in Fig. 3 that the amount of internal gating needed for the special transitions is always the same, being independent of n.

If the desired total count N does not happen to be exactly equal to one of the binomial coefficients $\binom{n}{2}$, the next larger value of n should be selected, and the resulting counter modified to cause it to skip a few states to reduce the count to N. If the proper states are skipped, the internal logic of the counter can remain simple, only minor changes being required. Table I shows how such modifications can be made to achieve any desired count $N=\binom{n}{2}-\delta$ (where $0\leq\delta\leq n-1$). This table lists the interstage transitions which are required of the register during the special transition mode, *i.e.*, whenever $x_1=1$. When $x_1=0$, the register shifts in the usual circulating fashion. For $\delta=0$ (first column), the pattern of transitions agrees with that already derived, as expressed in the above set of general equations: $1\rightarrow 2$, $2\rightarrow k$, $3\rightarrow 3$, etc. The transitions which must be changed if $\delta>0$ are indicated by the underlined entries in the other columns of the table. Note that these changes consist of a systematic permutation of two or three of the entries in the $\delta=0$ column. The pattern is clear for those intervening columns not shown explicitly.

TABLE I

STAGE-TO-STAGE TRANSITIONS REQUIRED WHEN $x_1=1$ TO ACHIEVE A COUNT $N=\binom{n}{2}-\delta$ IN A $2/n$ COUNTER. WHEN $x_1=0$, THE REGISTER SHOULD SHIFT IN THE USUAL MANNER

Source Stage		δ=0	1	2	3	···	$k-2$	$k-1$	k	$k+1$	$k+2$	···	$n-1$
1	→	2	2	2	2		2	2	2	2	2		2
2	→	k	1	k	k		k	$k+1$	1	$k+1$	$k+1$		$k+1$
3	→	3	3	n	3		3	3	3	n	3		3
4	→	4	4	4	$n-1$		4	4	4	4	$n-1$		4
⋮													
$k-2$	→	$k-2$	$k-2$	$k-2$	$k-2$		$k-2$	$k-2$	$k-2$	$k-2$	$k-2$		$k+2$
$k-1$	→	$k-1$	$k-1$	$k-1$	$k-1$		$k+1$	$k-1$	$k-1$	$k-1$	$k-1$		$k-1$
k	→	$k+1$	$k+1$	$k+1$	$k+1$		$k-1$	k	k	k	k		k
$k+1$	→	$k+2$	$k+2$	$k+2$	$k+2$		$k+2$	$k+2$	$k+2$	$k+2$	$k+2$		$k-2$
⋮													
$n-2$	→	$n-1$	$n-1$	$n-1$	4		$n-1$	$n-1$	$n-1$	$n-1$	4		$n-1$
$n-1$	→	n	n	3	n		n	n	n	3	n		n
n	→	1	k	1	1		1	1	$k+1$	1	1		1

The equations corresponding to the modified-transitions can be written down directly from the table; *e.g.*, for a 2/7 counter modified to count to $N=\binom{7}{2}-2=19$, we have $n=7$, $\delta=2$, $k=5$. The transitions required when $x_1=1$ are therefore

$$1\rightarrow 2$$
$$2\rightarrow 5$$
$$3\rightarrow 7$$
$$4\rightarrow 4$$
$$5\rightarrow 6$$
$$6\rightarrow 3$$
$$7\rightarrow 1$$

leading to the set of equations

$$x_1' = x_7$$
$$x_2' = x_1$$
$$x_3' = \bar{x}_1x_2 + x_1x_6$$
$$x_4' = \bar{x}_1x_3 + x_1x_4$$
$$x_5' = \bar{x}_1x_4 + x_1x_2$$
$$x_6' = x_5$$
$$x_7' = \bar{x}_1x_6 + x_1x_3.$$

The circuit is shown in Fig. 4.

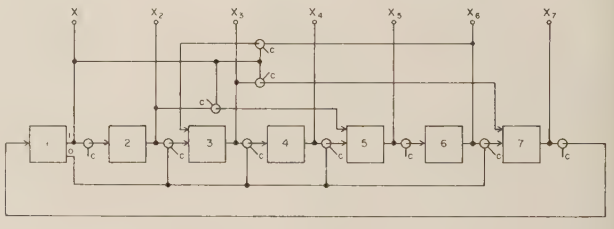


Fig. 4—A 2/7 counter modified to count to 19.

III. 3-OUT-OF-n COUNTERS

The generalization from $2/n$ to p/n counters ($p\geq 3$) will reduce the length of the register somewhat for a sufficiently large given value of N, but will increase the complexity of the logic circuitry, both for effecting the special transitions and for decoding the n register outputs into the $N=\binom{n}{p}$ state outputs. The search for simple transition logic is tantamount to a search for a systematic and simple way to progress through all $\binom{n}{p}$ states. We present one reasonable configuration for the $3/n$ counter. The minimization of p/n decoding trees is discussed in the next section.

The main change in going from the $2/n$ to $3/n$ case is the necessary use of two special transition modes instead of one. Whereas in the $2/n$ counter we had:

regular mode: $x_1 = 0$ pure shift

special mode: $x_1 = 1$ special transitions;

for the $3/n$ counter we have

regular mode: $g_n = 0$ pure shift

special mode A: $g_n\bar{x}_2 = 1$ special transitions

special mode B: $g_nx_2 = 1$ special transitions.

For $n\geq 7$, modes A and B are specified in Table II where

$$j = \left[\frac{n-1}{3}\right]$$
$$k = \left[\frac{j+3}{2}\right]$$

and

$$g_n = x_1\bar{x}_n\bar{x}_{n-1}\bar{x}_{n-2}\cdots\bar{x}_{n-j+1}.$$

TABLE II

SPECIAL STAGE-TO-STAGE TRANSITIONS REQUIRED IN A 3/n COUNTER

		Mode A	Mode B	
1	→	2	2	
2	→	$(j+2)$*	$j+2$	
3	→	3	3	
4	→	4	4	
.				
.				
.				
k	→	k	k	needed for $n \geq 10$ only
$k+1$	→	$k+1$	$k+2$	
$k+2$	→	$k+2$	$k+3$	
.				
.				
.				
j	→	j	$j+1$	
$j+1$	→	$j+1$	$n-j+1$	
$j+2$	→	$j+3$	$k+1$	
$j+3$	→	$j+4$	$j+3$	
.				
.				
.				
$n-j$	→	$n-j+1$	$n-j$	
$n-j+1$	→	$n-j+2$	$n-j+2$	
$n-j+2$	→	$n-j+3$	$n-j+3$	
.				
.				
.				
$n-1$	→	n	n	
n	→	1	1	

* Or any other convenient stage, since Mode A never occurs when $x_2=1$.

The equations corresponding to Table II may be written directly in the form:

$$x_1' = x_n$$
$$x_2' = x_1$$
$$x_3' = \bar{g}_n x_2 + g_n x_3$$
$$x_4' = \bar{g}_n x_3 + g_n x_4$$
$$\cdots$$
$$x_k' = \bar{g}_n x_{k-1} + g_n x_k$$
$$x_{k+1}' = \bar{g}_n x_k + g_n \bar{x}_2 x_{k+1} + g_n x_2 x_{j+2}$$
$$x_{k+2}' = (\bar{g}_n + x_2) x_{k+1} + g_n \bar{x}_2 x_{k+2}$$
$$x_{k+3}' = (\bar{g}_n + x_2) x_{k+2} + g_n \bar{x}_2 x_{k+3}$$
$$\cdots$$
$$x_{j+1}' = (\bar{g}_n + x_2) x_j + g_n \bar{x}_2 x_{j+1}$$
$$x_{j+2}' = \bar{g}_n x_{j+1} + g_n x_2$$
$$x_{j+3}' = (\bar{g}_n + \bar{x}_2) x_{j+2} + g_n x_2 x_{j+3}$$
$$x_{j+4}' = (\bar{g}_n + \bar{x}_2) x_{j+3} + g_n x_2 x_{j+4}$$
$$\cdots$$
$$x_{n-j}' = (\bar{g}_n + \bar{x}_2) x_{n-j-1} + g_n x_2 x_{n-j}$$
$$x_{n-j+1}' = (\bar{g}_n + \bar{x}_2) x_{n-j} + g_n x_2 x_{j+1}$$
$$x_{n-j+2}' = x_{n-j+1}$$
$$x_{n-j+3}' = x_{n-j+2}$$
$$\cdots$$
$$x_n' = x_{n-1}.$$

The general circuit is shown in Fig. 5. Note that the complexity of the transition logic now increases with n, but only in the number of inputs to the single large OR gate which forms $\bar{g}_n$.

For $n<7$, a degeneracy sets in, which renders Table II and these equations ambiguous. For $n=4$, pure circulation is adequate. For $n=5$ and 6, a single special mode suffices, and is employed under the condition $x_2x_1=1$: $3\rightarrow n$, $4\rightarrow 4$, $n-1\rightarrow n-1$.

As an example, consider a 3/10 counter, for which

$$N = \binom{10}{3} = 120$$
$$j = 3$$
$$k = 3.$$

The equations become:

$$x_1' = x_{10}$$
$$x_2' = x_1$$
$$x_3' = \bar{g}_{10} x_2 + g_{10} x_3$$
$$x_4' = \bar{g}_{10} x_3 + g_{10} \bar{x}_2 x_4 + g_{10} x_2 x_5$$
$$x_5' = \bar{g}_{10} x_4 + g_{10} x_2$$
$$x_6' = (\bar{g}_{10} + \bar{x}_2) x_5 + g_{10} x_2 x_6$$
$$x_7' = (\bar{g}_{10} + \bar{x}_2) x_6 + g_{10} x_2 x_7$$
$$x_8' = (\bar{g}_{10} + \bar{x}_2) x_7 + g_{10} x_2 x_4$$
$$x_9' = x_8$$
$$x_{10}' = x_9$$

where

$$g_{10} = x_1 \bar{x}_{10} \bar{x}_9 \bar{x}_8.$$

The circuit is shown in Fig. 6.

IV. p/n Decoding Trees

A p/n decoding tree is a combinational network that has n uncomplemented inputs $x_1, x_2, \cdots, x_n$, and $N = \binom{n}{p}$ outputs; each output has the binary value "1" for a particular and unique unordered p-tuple formed from the input set. In the formation of the outputs as switching functions of the inputs, all input combinations having just p uncomplemented variables cause one and only one output to have the value 1; all input combinations having less than p uncomplemented variables cause no outputs to have the value 1, and may be treated as "don't care" cases. It is therefore sufficient to form each output as a conjunction (product) of exactly p input variables, through one or more levels of AND gates. Examples are afforded by the 2/5 tree at the top of Fig. 2, and the 3/6 tree shown in Fig. 7. Note in the latter case that the outputs are generated by two levels of AND gates, the first of which forms certain pairs, and the second of which forms the $N=\binom{6}{3}=20$ triples, each from a first-level pair and a single variable.

It is of interest to find those classes of p/n decoding networks that have a cost that is minimal in some sense. If this cost is proportional to the total number of gates,

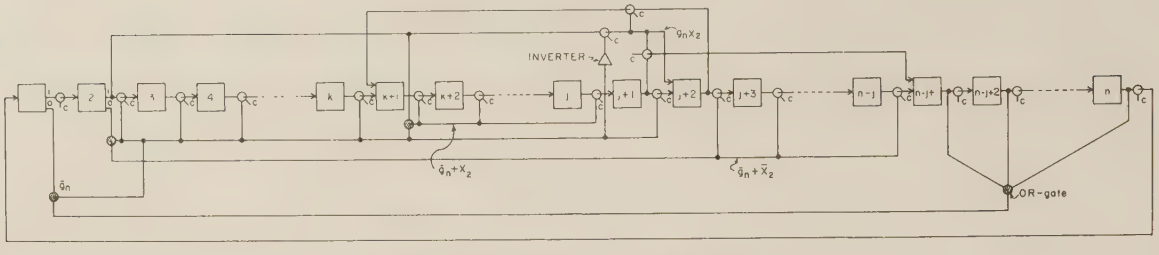


Fig. 5—General form of a $3/n$ ring counter.

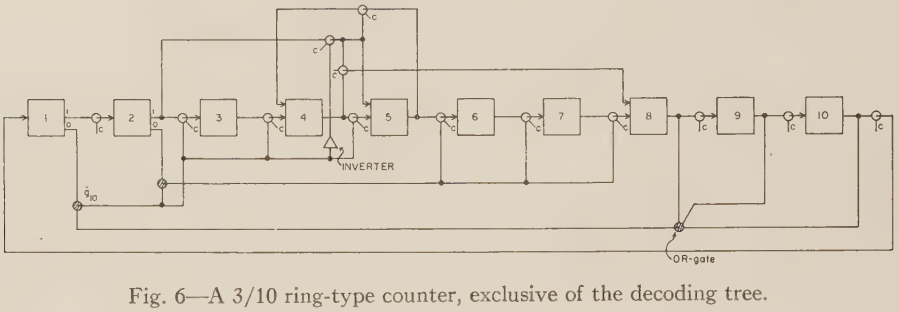


Fig. 6—A 3/10 ring-type counter, exclusive of the decoding tree.

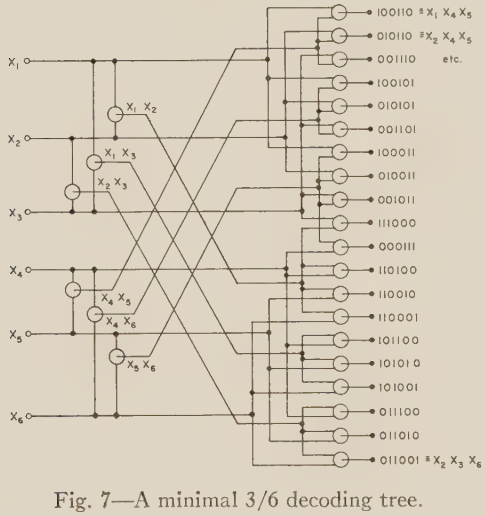


Fig. 7—A minimal 3/6 decoding tree.

regardless of the number of inputs per gate, then a single-level network may be used that consists of N p-input AND gates. Each gate in this configuration is driven by a different set of p input variables. Since each of the N outputs must be formed by some AND gate, no fewer than N gates will suffice, so such a network is minimal.

A more realistic cost figure is one that weights AND-gate *inputs* equally. Let $C_p(n)$ denote the minimum total number of AND-gate inputs required for a full p/n tree. For $p=2$, the minimum-gate realization is also the realization with minimum gate inputs, so

$$C_2(n) = 2\binom{n}{2} = n(n-1).$$

For $p=3$, a rule about to be described yields (for $n>3$)

$$C_3(n) = 2\binom{n}{3} + 2\binom{r}{2} + 2\binom{n-r}{2}$$

where

$$r = \left[\frac{n}{2}\right].$$

(As before, the brackets indicate the integral part of the quantity within the brackets.) This expression becomes

$$C_3(n) = \tfrac{1}{6}n(n-2)(2n+1)$$

for n even, and

$$C_3(n) = \tfrac{1}{6}(n+1)(n-1)(2n-3)$$

for n odd.

The formation of the net requires first that the input variables be divided into two groups of equal (or almost equal) size—say, r and $(n-r)$—and that all pairs be formed for each of these groups separately, just as if they were two trees of type $2/r$ and $2/(n-r)$. Each of the $3/n$ outputs is then formed by combining one of these pairs with one of the input variables. (See Fig. 7.) Since every possible triple of input variables will have at least two of its three variables in one group or the other, every output can be formed in the second level with an AND gate having only two inputs. This second level accounts for the first term in the expression for $C_3(n)$, and the two smaller trees in the first level account for the second and third terms.

The proof that no fewer number of pairs will suffice in the first level is presented in Appendix I. It can also be shown (for $n>3$) that the total number of gate inputs in the $3/n$ network cannot be reduced below the value given above as $C_3(n)$ by allowing one or more of the outputs to be formed directly as a conjunction of three input variables. Since the proof is tedious and not very enlightening, it is not included here.

For $p=4$, a two-level network in which each output is formed from a "pair of pairs" (such as $[(x_1x_2)(x_3x_4)]$)

ll be shown to give the upper bound

$$C_4(n) \leq 2\binom{n}{4} + n(n-3).$$

his bound is probably actually achieved for $n>6$. It nnot be reduced within the class of pair-of-pairs netrks, but a proof is lacking that the pair-of-pairs netrk form is the least costly form. For $n\leq 6$, it can cerinly be improved. In the $n=6$ case, we may form the irs

$$(x_1x_2) \qquad (x_4x_5)$$
$$(x_1x_3) \qquad (x_4x_6)$$
$$(x_2x_3) \qquad (x_5x_6),$$

en, in a second level, the triples

$$[(x_1x_2)x_3] \qquad [(x_4x_5)x_6],$$

d finally, all of the outputs in a third level, to give a st $C_4(6)=46$ instead of 48. Similarly, the pairs (x_1x_2) d (x_3x_4) give $C_4(5)=18$ instead of 20. Obviously, (4) = 4 instead of 5.

The pair-of-pairs bound given above may be arrived as follows. Consider all those quadruples containing me one variable, say, x_i. Clearly, one of the constitut pairs of every such quadruple must contain x_i. To rm all these quadruples, x_i must, therefore, occur in a ir with every one of the other variables, except, at ost, two, and these two must of necessity be paired tother. Thus x_i is involved in $(n-3)$ pairs. This is true each of the n variables, so that the total number of irs needed is

$$\tfrac{1}{2}n(n-3).$$

he factor $\frac{1}{2}$ arises because each pair has been counted vice, once for each of its constituent variables. Each ir, and each of the $N=\binom{n}{4}$ outputs, requires a twoput AND gate, from which facts the above bound on $_4(n)$ follows directly.

The matter of which pairs should be formed in the rst level of a pair-of-pairs network may be settled by e following inductive procedure. This procedure deribes how to form the new pairs in a network with +1 input variables, starting with a network having n put variables.

1) Pick two variables (say, x_i and x_j) that are not paired in the n-variable network, and form a pair (x_ix_j) with them.
2) Form pairs between the new variable x_{n+1} and all other variables *except* x_i and x_j.

his process adds a total of $(n-1)$ pairs to the original $(n-3)$ pairs, to give in the $(n+1)$-variable network a tal of

$$\tfrac{1}{2}n(n-3) + (n-1) = \tfrac{1}{2}(n+1)(n-2)$$

irs, as it should.

It is also relatively simple to show that all new quadples containing x_{n+1} may now be formed from a pair pairs. Let x denote a generic variable other than x_i, x_j or x_{n+1}. Then the pair (x_ix_j) and all pairs of the form $(x_{n+1}x)$ are available from construction in steps 1) and 2). The completeness of the n-variable network by itself assures us that every x in quadruples such as $[x_ix_jxx]$ is already paired with either x_i or x_j. Thus, new quadruples of the form $[x_{n+1}x_ix_jx]$, $[x_{n+1}x_ixx]$, and $[x_{n+1}x_jxx]$ are certainly decomposable into available pairs. The only remaining possibility, $[x_{n+1}xxx]$, is also decomposable, since every triple of x's must have been covered by at least one pair in the original n-variable network.

Fig. 8 shows the first level of a 4/7 pair-of-pairs decoding tree.

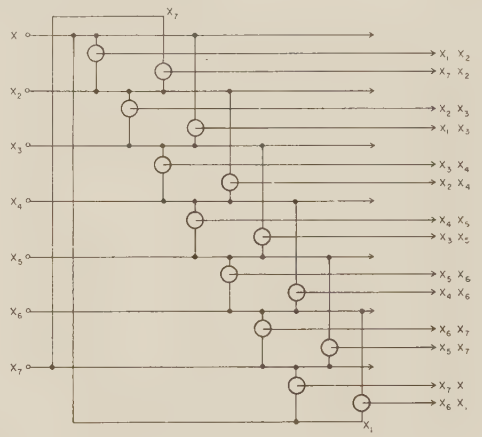


Fig. 8—First level of a 4/7 pair-of-pairs decoding tree.

For $p>4$, no systematic decoding trees have been derived, let alone values of the minimal cost $C_p(n)$. The cases already considered suggest that the cost need not increase more rapidly than $2\binom{n}{p}$ (*i.e.*, as $2n^p/p!$) as n increases with p fixed, but that a number of levels greater than two will probably be necessary if $p>4$.

Thus, the general theoretical problem of forming a minimal realization of the p/n decoding tree remains unsolved for $p>4$, and the solution is incomplete for $p=4$. Despite superficial similarities to conventional decoding trees (those with n inputs and 2^n outputs),[3] p/n trees appear to be basically different, and no extension or modifications of the minimality proofs for conventional trees could be found to apply to the present problem.

V. An Application to Multiple-Coincidence Magnetic Memories

The p/n counters under discussion also offer an alternate mode of driving multiple-coincidence magnetic memory arrays which permits a larger size memory for the same number of drivers, at the expense of a somewhat more complicated address code.[4,5] In the usual

[3] A. W. Burks, *et al.*, "Complete decoding nets: general theory and minimality," *J. Soc. Ind. and Appl. Math.*, vol. 2, pp. 201–243; December, 1954.

[4] R. L. Ashenhurst and R. C. Minnick, "Multiple-coincidence magnetic storage systems," in "Theory of Switching," Harvard Computation Lab., Cambridge, Mass., Rept. BL-4, pp. VI-41–VI-44; July, 1953.

[5] N. Blachman, "On the wiring of two-dimensional multiple-coincidence magnetic memories," IRE Trans. on Electronic Computers, vol. EC-5, pp. 19–21; March, 1956.

configuration of a memory as in Fig. 9, address counters A_1 and A_2 feed decoding trees D_1 and D_2, respectively, which, in turn, drive simultaneously one horizontal and one vertical line of the memory. The coincidence of signals on these two lines causes selection at the intersecting co-ordinate point in the array.[6]

Thus, a square array with a total of M drive lines provides for $M^2/4$ intersection points. These lines are separated into two groups of $M/2$ each, so arranged that only one line in each group is driven at any one time. Actually, however, two lines could be selected out of a total of M lines in $M^2/2-M/2$ ways, not $M^2/4$ ways. This number is achieved in the rearrangement shown in Fig. 10, in which this larger number of intersecting points is placed in a triangular rather than a square array.[7] The two counters and trees are merged into a single counter A_0 and tree D_0, arranged so that the tree drives the array simultaneously on just two lines. The array is then threaded so that each line passes through exactly $M-1$ points, intersecting each of the other $M-1$ lines just once. In short, line i passes through every point whose horizontal *or* vertical co-ordinate is i, relative to an origin of co-ordinates at the point O. In this way, the number of intersection points in the memory array may be almost doubled for the same number of drivers.

Various types of counters A_1 and A_2 can be employed in Fig. 9, provided the proper decoding trees D_1 and D_2 are used with them to yield a one-line-at-a-time output. For example, a standard binary counter can be associated with a conventional decoding tree; a ring counter can be used with no tree at all; or a $2/n$ counter can drive a $2/n$ decoding tree. Similarly, a choice of counters and trees can be made in Fig. 10 for A_0 and D_0, provided only that D_0 has a two-line-at-a-time output. Probably the simplest arrangement here is a $2/M$ counter for A_0, with direct-through lines (no tree at all) for D_0.

If triple-coincidence rather than double-coincidence selection is employed, the arrangements of Figs. 9 and 10 may be directly extended to three dimensions. In the standard structure of Fig. 11, three sets of drive lines having $M/3$ lines each, with supporting counters and trees, provide for a cubical array with $M^3/27$ intersection points. Each drive line now drives a *plane* instead of a *line* of points. Corresponding to the $2/M$ triangular array of Fig. 10, the $3/M$ array of Fig. 12 has the shape

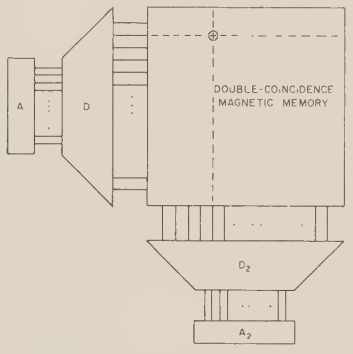


Fig. 9—Double-coincidence magnetic memory with standard access to a square array.

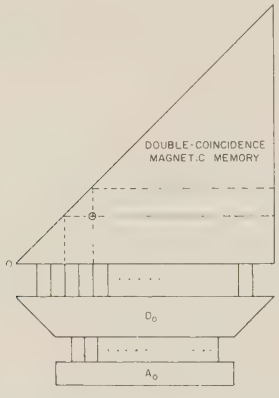


Fig. 10—Double-coincidence magnetic memory with suggested two-line-at-a-time access to a triangular array.

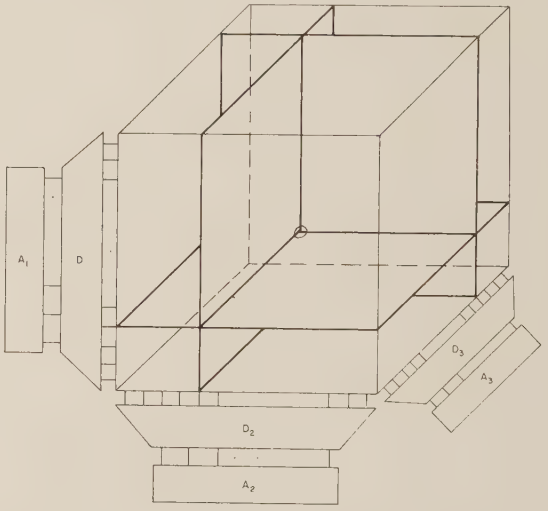


Fig. 11—Triple-coincidence magnetic memory with standard three-dimensional access to a cubical array.

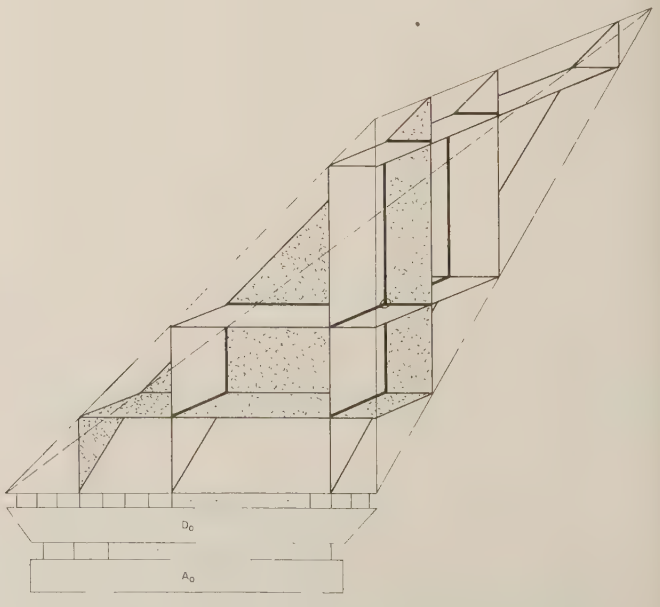


Fig. 12—Triple-coincidence magnetic memory with suggested three-line-at-a-time access to a tetrahedronal array.

[6] Actually, each intersection point or "core" may consist of several cores driven simultaneously, and intended for parallel access.

[7] This two-dimensional configuration was described by Ashenhurst and Minnick in their study of multiply-threaded magnetic core arrays.[4]

of a tetrahedron. Each of the selecting planes drives $\binom{M-1}{2}$ points—in fact, drive line i drives plane i, which drives all points, one of whose three co-ordinates is i. Note that whereas line i in Fig. 10 was reflected off of the hypotenuse of the triangle, plane i in Fig. 12 is involved in a double reflection off of the rear faces of the tetrahedron. A single counter-tree pair now provides three-line-at-a-time drive to select any one of the $\binom{M}{3} = M^3/6 - M^2/2 + M/3$ cores. Thus, for the same number M of drivers, the tetrahedronal array has up to $27/6 = 4.5$ times as many intersection points as the cubical array.

VI. Comparison of Ring-Type and Other Counters

Constant-weight ring-type counters can be compared with one another and with other kinds of counters on the basis of the number of logical and storage elements required for a given desired count N. We take the cost function to be the number of gate inputs (as in the discussion of p/n trees in Section IV), which is equal to the number of rectifiers if these gates are realized with rectifiers according to standard practice. Allowance can be made for the cost of the storage elements in the circuit through a ratio β:

$$\beta = \frac{\text{cost of one register stage (including transfer circuitry)}}{\text{cost of one gate input (\textit{e.g.}, one rectifier)}}.$$

In practice, a value of β somewhere between 5 and 20 might be expected. Through this ratio, the equivalent number of rectifiers $R_p(N)$ may be compared for various types of p/n counters.

For a $1/n$ counter, there are no rectifiers, so the equivalent cost is

$$R_1(N) = \beta n = \beta N.$$

For a $2/n$ counter, all of the gatery is confined to the decoding tree, except for two two-input AND gates required for the special transitions:

$$R_2(N) = \beta n_2 + 4 + C_2(n_2)$$
$$= \beta n_2 + 4 + 2N$$

for the case when $N = \binom{n_2}{2}$. If N is not exactly equal to one of these binomial coefficients, we must take the next largest integer value of n_2. A few more gates may be required to effect the transitions indicated in Table I, but the expression for $R_2(N)$ is still very nearly correct.

For a $3/n$ counter, the extra gatery requires about

$$10 + j + 1 = 11 + \left\lceil \frac{n-1}{3} \right\rceil$$

rectifiers so

$$R_3(N) = \beta n_3 + 11 + \left\lceil \frac{n_3 - 1}{3} \right\rceil + C_3(n_3)$$

$$R_3(N) = \beta n_3 + 11 + \left\lceil \frac{n_3 - 1}{3} \right\rceil + 2N$$
$$+ 2\binom{r_3}{2} + 2\binom{n_3 - r_3}{2}$$

where n_3 is the smallest integer such that $N \leq \binom{n_3}{2}$, and $r_3 = \lceil n_3/2 \rceil$. Again, reduction of the count below $\binom{n_3}{3}$ may require a few more gates, which we choose to neglect.

These formulas are compared graphically in Fig. 13, which displays those regions of the plot with axes N and β in which each ring-type counter has the least cost $R_p(N)$. For this plot, we may conclude directly that for large N: 1) pure $(1/n)$ ring counters are optimal only for values of β less than about 2; 2) $3/n$ counters are optimal only for values of β greater than 8 or 10; and 3) $2/n$ counters are cheapest in the intermediate range of β. For small N, $3/n$ counters are relatively economical only for large β, and then only for $N > 16$, and the $1/n$ variety compares favorably for increasingly larger values of β. Again, $2/n$ counters are optimal in the intermediate region.

It is worthwhile to compare these ring-type counters with conventional binary counters, constructed from complementing-input (trigger) flip-flops and augmented with the necessary portion of a complete decoding tree. Fig. 14 shows such a configuration, depicting one of several possible interstage transfer arrangements. (The results to be derived do not depend on which transfer arrangement is used.) The equivalent cost is

$$R_b(N) = \beta n_b + c_b(N).$$

Here, n_b is the smallest integer such that $N \leq 2^{n_b}$; that is

$$n_b = 1 + [\log_2 (N - 1)].$$

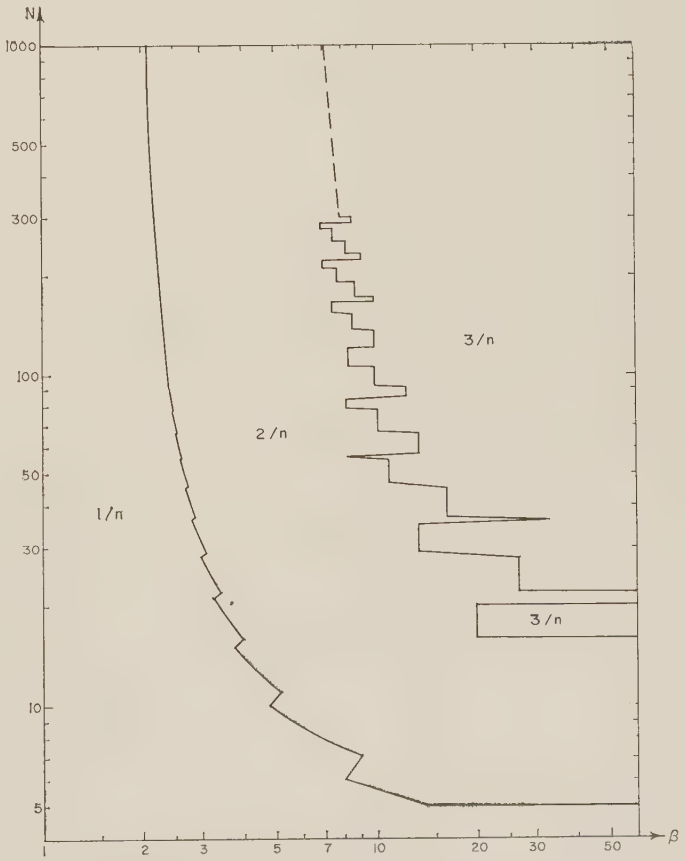


Fig. 13—Chart for the selection of the cheapest ring-type counter for given count N and register stage cost β.

The cost, $c_b(N)$, of an incomplete decoding tree is derived in Appendix III. For simplicity, we neglect the cost of the single AND gate required to reset the binary counter to $000 \cdots 0$ after a count of N is reached. Also, to establish a reasonable basis of comparison, we assume that one binary counter stage has the same cost as one stage of the shift register used in the p/n counters.

Fig. 15 extends the previous comparison to include the binary case, as well as an "inverted feedback" counter (IFB) to be described below. These curves and the cost formulas derived above reveal the following information.

1) For large N, binary and $3/n$ counters cost about the same, the binary form being only slightly preferable. Thus, the $3/n$ form is never actually the cheapest form, and for this reason does not appear on the figure.

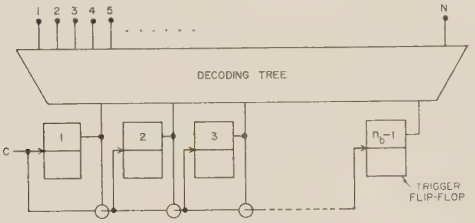


Fig. 14—Conventional binary counter augmented with an incomplete decoding tree to provide individual state outputs.

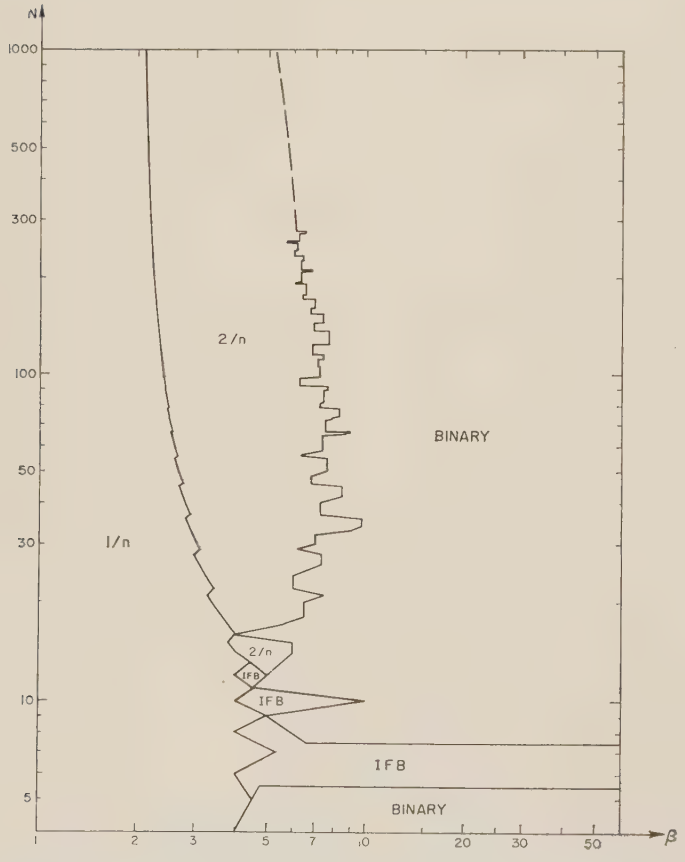


Fig. 15—Chart for the selection of the cheapest counter of count N and stage cost β, among ring-type counters, the binary counter with decoding tree, and the inverted feedback (IFB) counter with decoding tree.

2) For large N, the range of optimality of $2/n$ counters is about the same as before, the range of β being only slightly narrowed.
3) For $\beta > 10$, the binary form is always optimal (except for $N = 6$ and 7, when the IFB is a little less costly).
4) For $\beta < 2$, the $1/n$ form is always optimal.

Actually, these curves are not intended to be very accurate for small N, in view of the approximations made in the expressions for the cost functions $R(N)$. Nevertheless, the above results are certainly at least qualitatively valid in this region.

It should be emphasized that Figs. 13 and 14 and these conclusions apply only when all N-state outputs of the tree are needed. If a lesser number of outputs are adequate, the determination of the optimum p for given N and β can be carried out with the above expressions for $R_p(N)$, by reducing the magnitude of the $C_p(n)$ or $c_b(N)$ term to correspond with the smaller size of the decoding tree.

For small values of N, a few other configurations based on the shift register offer competitive alternatives to the $2/n$ and binary counters. A simple circulating register with inversion in the feedback loop [Fig. 16(a)] gives $N = 2n$ states, starting at $00 \cdots 00$, and the decoding tree requires only $2N = 4n$ rectifiers. For example, for $n = 5$, the sequence of states is:

x_5	x_4	x_3	x_2	x_1	Output
0	0	0	0	0	$\bar{x}_1\bar{x}_5$
0	0	0	0	1	$\bar{x}_2 x_1$
0	0	0	1	1	$\bar{x}_3 x_2$
0	0	1	1	1	$\bar{x}_4 x_3$
0	1	1	1	1	$\bar{x}_5 x_4$
1	1	1	1	1	$x_1 x_5$
1	1	1	1	0	$x_2\bar{x}_1$
1	1	1	0	0	$x_3\bar{x}_2$
1	1	0	0	0	$x_4\bar{x}_3$
1	0	0	0	0	$x_5\bar{x}_4$
(0	0	0	0	0)	

The right-hand column of this list indicates the stages of the register that should be AND-gated to provide the $N = 10$ state outputs.

When N is odd, it is necessary to use an $(N+1)/2$ stage register, with the feedback signal generated from

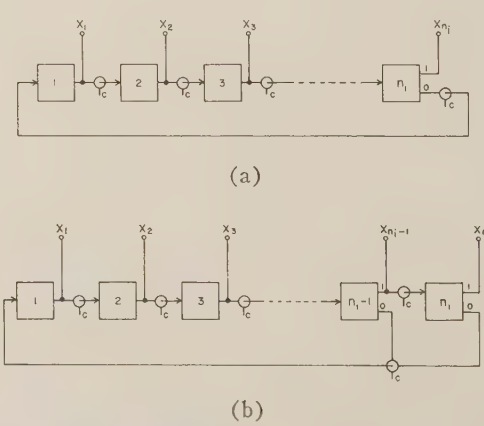


Fig. 16—Inverted feedback counter. (a) N even. (b) N odd;

$$n_1 = \left[\frac{N+1}{2}\right].$$

the last two stages as shown in Fig. 16(b). This connection causes state $111 \cdots 1$ to be skipped. The equivalent cost for both even and odd N is, therefore,

$$R_i(N) = (\beta + 4)\left[\frac{N+1}{2}\right].$$

The virtues of the *feedback shift register* of Fig. 17 have been known for some time.[8,9] In short, any count $N \leq 2^n$ can be achieved with suitable logic in the feedback network. For many values of n, a count of $N = 2^n - 1$ can be achieved with only a single two-input exclusive-OR gate in this network, driven by the last stage and one other stage. In general, an incomplete decoding tree similar to the type used above with the binary counter will be required to provide the N-state outputs. The relationship between the logical structure of the feedback network and the possible values of N has been investigated. Although the problem is not completely solved, a partial understanding has been achieved, as well as complete solutions in some special cases, *e.g.*, when the network can be constructed entirely from exclusive-OR gates and inverters.

VII. Digit-Weighted p/n Counting Codes

One desirable characteristic of a counting code is the *digit-weighted* feature, with which it is possible to associate with each binary digit x_i a *weight* α_i, such that the numbers

$$\sum_{i=1}^{n} \alpha_i x_i$$

form a sequence of uniformly-spaced values as the counter advances through its permissible states. The conventional binary code has this digit-weighted feature, the weights being simply $\alpha_i = 2^{i-1}$; the resulting sequence runs through the integer from 0 to $2^n - 1$.

It is shown in Appendix II that $2/n$, $3/n$, and $4/n$ codes cannot be digit-weighted, except for a few degenerate cases. This is true regardless of the order of the states in the counting sequence. These exceptional cases are as follows:

p	Digit-Weighted Codes	Possible Weights
1	$1/n$	$0, 1, 2, \cdots, n-1$
2	2/3	0, 1, 2
	2/4	0, 1, 2, 4 0, 2, 3, 4
3	3/4	0, 1, 2, 3
	3/6	0, 3, 5, 6, 7, 14 0, 7, 8, 9, 11, 14
4	4/5	0, 1, 2, 3, 4

[8] N. Zierler, "Several Binary Sequence Generators," Lincoln Lab., Mass. Inst. Tech., Cambridge, Tech. Rept. 95; September 12, 1955.

[9] B. Elspas, "Theory of autonomous linear sequential networks," IRE Trans. on Circuit Theory, vol. CT-6, pp. 45–60; March, 1959.

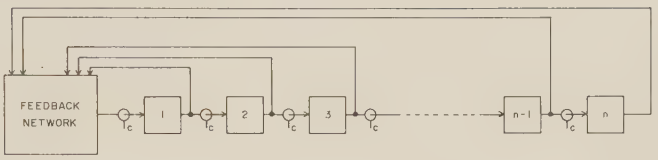


Fig. 17—Feedback shift register.

VIII. Conclusion

The study of digital counters from which a separate output is desired for each counting state has provided some useful results for design purposes, and has raised several interesting theoretical questions as well. Constant-weight counters of the type described actually have been constructed or simulated in the laboratory for several values of n and p, and show possible application to other situations, such as the access circuitry for coincident-line memory arrays.

The comparison of these constant-weight counters with one another and with conventional counters reveals many situations when the constant-weight form is more economical from an engineering standpoint. Depending on the relative cost of logic and binary storage in a circuit, and on how many outputs are needed, they may require considerably fewer circuit elements. Even without state outputs, $2/n$ counters may be preferred to binary counters because their register-like structure is sometimes easier to realize than a chain of flip-flops with complementing inputs, *e.g.*, in a magnetic-core circuit. Also, the fact that the number of 1's remains constant indicates that a constant load is presented to the power and pulse supplies. This is frequently an advantage, particularly in magnetic circuits.

While the design problems for counters and trees likely to be used in practice have been solved, a number of theoretical questions remain unanswered. Simple counting sequences for $p \geq 4$ are still lacking, and there may even be an improved version in the $p = 3$ case. Minimization of p/n decoding trees is unsolved for $p > 3$, although a minimal version of one network form (the pair-of-pairs tree) has been obtained for $p = 4$. This form of the $4/n$ tree is conjectured to be minimal among all possible forms, but a proof of this conjecture is lacking. Finally, a simpler and more complete proof of the impossibility of weighted p/n codes would be desirable.

Appendix I

Minimality of the $3/n$ Decoding Tree[10]

It will now be proved that the value of $C_3(n)$ presented in Section IV, and shown there to be an upper bound on the number of AND-gate inputs in a $3/n$ decoding tree, is also a lower bound. This will be demonstrated

[10] The essence of this proof was developed by Prof. S. Feferman of Stanford University, presently at the Institute for Advanced Study, Princeton, N. J., who participated in the investigation of minimal p/n trees, and whose contributions to this study are gratefully acknowledged. Prof. Feferman also provided the aforementioned proof that the use of three-input gates cannot reduce the cost of the $3/n$ tree.

inductively by showing that the number of first-level pairs $P(n+1)$ that must be formed in the $(n+1)$-case exceeds the number $P(n)$ that must be formed in the n-case by at least $r=[n/2]$; *i.e.*,

$$P(n+1) \geq r + P(n).$$

If this can be shown, then $P(n)$ can be bounded according to:

$$P(n+1) \geq \left[\frac{n}{2}\right] + \left[\frac{n-1}{2}\right] + \left[\frac{n-2}{2}\right] + \cdots + \left[\frac{4}{2}\right] + P(4).$$

Since two pairs are needed for $n=4$, $P(4)=2$. Summing this series gives

$$P(n) \geq \binom{r}{2} + \binom{n-r}{2}$$

from which the expression for $C_3(n)$ follows directly.

To show that at least r new first-level pairs are needed for each increase of n by *one*, we need only show that at least one input variable must occur in at least $r=[n/2]$ first-level pairs in an $(n+1)$-input tree. (For, by relabeling the variables, this input variable can always be made to be x_{n+1}.) This will also be shown by induction on n, as follows.

1) When $n+1=4$, then $r=1$, and every variable occurs once in first-level pairs of the four-input tree. So the statement is true for $n=3$.
2) By the induction hypothesis, some variable (say, x_i) occurs at least $[(n-1)/2]$ times in first-level pairs of the n-input tree. We need to show that x_i (or x_{n+1}) occurs at least $[n/2]$ times in first-level pairs of the $(n+1)$-input tree.
   a) *n odd:*

$$\left[\frac{n-1}{2}\right] = \left[\frac{n}{2}\right],$$

   so the statement is directly true.
   b) *n even:*

$$\left[\frac{n-1}{2}\right] = \left[\frac{n}{2}\right] - 1.$$

   If x_{n+1} occurs $[n/2]$ times, the statement is true immediately. If not, it will be shown that pair $(x_i x_{n+1})$ is present, the inclusion of which increases the occurrences of x_i from $[n/2]-1$ to $[n/2]$. If $(x_i x_{n+1})$ were not present, we could certainly find some other variable (x_j, say) that is not paired with either x_i or x_{n+1}, for the total number of pairs containing either x_i or x_{n+1} is no greater than

$$2\left[\frac{n-1}{2}\right] = n-2,$$

   whereas every variable x_j is potentially pairable with all n variables besides itself. Therefore, the absence of all three of the pairs $(x_i x_{n+1})$, $(x_i x_j)$, and $(x_j x_{n+1})$ would leave the triple $(x_i x_j x_{n+1})$ uncovered by any first-level pair, contrary to hypothesis. So the pair $(x_i x_{n+1})$ must be present; x_i then occurs in pairs the requisite number of times, and the statement is true.

Thus, increasing n to $n+1$ requires an increase in the number of pairs by at least r, proving the assertion, and establishing the validity of the expression for $C_3(n)$ as a lower bound.

Appendix II

Nonexistence of Digit-Weighted p/n Counting Codes

We now prove that no $2/n$, $3/n$, or $4/n$ codes can be digit-weighted, except for a few degenerate cases.

We first observe that the values in the sequence of numbers

$$\sum_{i=1}^{n} \alpha_i x_i$$

can be taken to be integers, with no loss of generality; for, if the spacing between successive values is s instead of 1, replace each α_i by α_i/s, and if the resulting values contain a fractional part $f \neq 0$, replace each resulting α_i by $\alpha_i - f/p$.

We now seek a set of numbers α_i $(i=1, 2, \cdots, n)$ such that every sum of exactly p of the α_i yields a unique integer in a continuous sequence of $\binom{n}{p}$ integers. Let the weights α_i be ordered in a decreasing sequence, $\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_n$. Clearly, no two α_i can be equal without duplication of several of the sums. For the $2/n$ case, the first few sums of pairs can be ranked, and set equal to the largest integer (N, say) of the sequence, the second largest, the third largest, etc.

	Case A		*Case B*	
$\alpha_1 + \alpha_2 = N$				
$\alpha_1 + \alpha_3 = N-1$				
$\alpha_2 + \alpha_3 = N-2$	or	$N-3$		
$\alpha_1 + \alpha_4 = N-3$	or	$N-2$,	respectively.	

These conditions allow α_1, α_3, and α_4 to be expressed in terms of α_2:

Case A	*Case B*
$\alpha_1 = \alpha_2 + 1$	$\alpha_1 = \alpha_2 + 2$
$\alpha_2 = \alpha_2$	$\alpha_2 = \alpha_2$
$\alpha_3 = \alpha_2 - 1$	$\alpha_3 = \alpha_2 - 1$
$\alpha_4 = \alpha_2 - 3$	$\alpha_4 = \alpha_2 - 2.$

In both cases, the remaining sums involving α_1 through α_4 automatically fill in the next two numbers of the sequence:

$$\alpha_2 + \alpha_4 = N - 4$$
$$\alpha_3 + \alpha_4 = N - 5.$$

This set of weights is now satisfactory for $n=4$ (*e.g.*, 0, 1, 2, 4, or 0, 2, 3, 4). The next weight α_5 must now be chosen to give

$$\alpha_1 + \alpha_5 = N - 6,$$

namely, $\alpha_5=\alpha_2-6$. There now appears in the sequence a gap:

Case A	*Case B*
$\alpha_1 + \alpha_5 = N - 6$	$\alpha_1 + \alpha_5 = N - 6$
$\alpha_2 + \alpha_5 = N - 7$	$\alpha_2 + \alpha_5 = N - 8$
$\alpha_3 + \alpha_5 = N - 8$	$\alpha_3 + \alpha_5 = N - 9$
$\alpha_4 + \alpha_5 = N - 10$	$\alpha_4 + \alpha_5 = N - 10.$

This gap dictates the value of α_6:

$\alpha_1 + \alpha_6 = N - 9$ or	$\alpha_1 + \alpha_6 = N - 7$
i.e.,	*i.e.*,
$\alpha_6 = \alpha_2 - 9$	$\alpha_6 = \alpha_2 - 7.$

But now duplication of sums results in both cases, since

$$\alpha_2 + \alpha_6 = \alpha_4 + \alpha_5 \qquad \alpha_2 + \alpha_6 = \alpha_3 + \alpha_5.$$

Thus, for $n>4$, no set of weights α_i exists such that their pair-wise sums form a sequence of successive integers. Consequently, no $2/n$ counting codes except the $2/3$ (*e.g.*, weights 0, 1, 2), and the $2/4$ cases can be digit-weighted.

A similar proof shows the nonexistence of digit-weighted $3/n$ codes for $n>6$. Just as before, the largest sums are:

	Case A	*Case B*	
$\alpha_1 + \alpha_2 + \alpha_3 = N$			
$\alpha_1 + \alpha_2 + \alpha_4 = N - 1$			
$\alpha_1 + \alpha_3 + \alpha_4 = N - 2$	or	$N - 3$	
$\alpha_1 + \alpha_2 + \alpha_5 = N - 3$	or	$N - 2,$	respectively.

All other possibilities such as $\alpha_2+\alpha_3+\alpha_4=N-3$ for the fourth of these conditions are easily ruled out, from the consequence that they lead directly to duplicate sums such as

$$\alpha_1 + \alpha_4 + \alpha_5 = \alpha_2 + \alpha_3 + \alpha_5.$$

The next two values in the sequence follow automatically for both cases:

$$\alpha_1 + \alpha_3 + \alpha_5 = N - 4$$
$$\alpha_1 + \alpha_4 + \alpha_5 = N - 5.$$

The remaining four triples involving α_1 through α_5, expressed in terms of α_3, reveal a sequence of integers with a gap in it.

Case A		*Case B*
$\alpha_2 + \alpha_3 + \alpha_4 = 3\alpha_3$	or	$3\alpha_3 = 1$
$\alpha_2 + \alpha_3 + \alpha_5 = 3\alpha_3 - 2$		$3\alpha_3$
$\alpha_2 + \alpha_4 + \alpha_5 = 3\alpha_3 - 3$		$3\alpha - 1$
$\alpha_3 + \alpha_4 + \alpha_5 = 3\alpha_3 - 4$		$3\alpha_3 - 3$

so that weighting of a $3/5$ code is impossible. Triple sums involving α_6 produce an additional sequence of values with a gap.

Case A		*Case B*
$\alpha_1 + \alpha_2 + \alpha_6 = M$	or	M
$\alpha_1 + \alpha_3 + \alpha_6 = M - 1$	or	$M - 2$
$\alpha_1 + \alpha_4 + \alpha_6 = M - 2$	or	$M - 3$
$\alpha_1 + \alpha_5 + \alpha_6 = M - 4$	or	$M - 4$

where $M=N-(\alpha_3-\alpha_6)$.

These last two sequences can be merged so that the gaps mesh properly in just one way for each case:

Case A	*Case B*
$3\alpha_3 = N - 9$	$3\alpha_3 + 1 = N - 6$
$M = N - 6$	$M = N - 9$

completing the sequence from $N-6$ to $N-13$ with no gaps. The remaining triple sums involving α_6 now provide the next 6 values automatically for both cases, yielding a valid weighting for the $3/6$ code:

$$\alpha_2 + \alpha_3 + \alpha_6 = N - 14$$
$$\alpha_2 + \alpha_4 + \alpha_6 = N - 15$$
$$\left.\begin{aligned}\alpha_3 + \alpha_4 + \alpha_6 = \\ \alpha_2 + \alpha_5 + \alpha_6 = \end{aligned}\right\} \begin{cases} N - 16 \\ N - 17 \end{cases}$$
$$\alpha_3 + \alpha_5 + \alpha_6 = N - 18$$
$$\alpha_4 + \alpha_5 + \alpha_6 = N - 19.$$

For example, the set of weights 0, 3, 5, 6, 7, 14 or 0, 7, 8, 9, 11, 14 constitutes a satisfactory set. If we attempt to continue the sequence, however, the next condition that must be satisfied, namely,

$$\alpha_1 + \alpha_2 + \alpha_7 = N - 20$$

results in a sequence from $N-20$ to $N-31$, but with $N-23$ (Case A) or $N-21$ (Case B) missing. Clearly, no choice of α_8 can fill this gap without duplication. Therefore, no digit-weighting of $3/n$ counting codes exists except for the $3/4$ code (*e.g.*, with weights 0, 1, 2, 3) and the $3/6$ code.

An almost identical proof also shows that $4/n$ codes cannot be digit-weighted, except for the trivial $4/5$ case (*e.g.*, weights 0, 1, 2, 3, 4). For p/n codes with $p>4$, the same technique of proof becomes very intricate, but it appears very unlikely that any additional digit-weightings are possible, aside from the $(n-1)/n$ cases. Also, since any valid weighting for a p/n code would also be valid for an $(n-p)/n$ code, all codes of the types $(n-2)/n$, $(n-3)/n$, and $(n-4)/n$ cannot be digit-weighted, except for the degenerate cases already mentioned; *e.g.*, no $5/9$ weighting exists.

Another argument excludes a large number of the remaining cases. Adding together all $\binom{n}{p}$ expressions of the form

$$\sum_{\substack{\text{exactly}\\ p \text{ terms}}} \alpha_i = N - j + 1 \qquad \left[j = 1, 2, \cdots \binom{n}{p} \right]$$

we obtain

$$\binom{n-1}{p-1} \sum_{i=1}^{n} \alpha_i = \sum_{j=1}^{\binom{n}{p}} (N - j + 1)$$

$$= \binom{n}{p}(N+1) - \left[\frac{\binom{n}{p}+1}{2} \right].$$

The coefficient of the left-hand summation is the number of p-tuples containing any given α_i and arises from the observation that, once this α_i is selected, the other $p-1$ α's in the p-tuple must be selected from the remaining set of $n-1$ α's. This expression reduces to

$$\sum_{i=1}^{n} \alpha_i = \frac{n}{2p}\left\{2N + 1 - \binom{n}{p}\right\}.$$

It is not difficult to show that all of the α_i may be assumed to be integers, with no loss of generality, so that any selection of values for n and p which renders the right-hand side of this equation nonintegral is not an acceptable selection for a weighted p/n code. For example, for $p=5$, $n=11$,

$$\sum_{i=1}^{11} \alpha_i = \frac{11}{10}\left\{2N + 1 - \binom{11}{5}\right\}$$

$$= \frac{11}{10}(2N - 461) \neq \text{an integer}.$$

Therefore, no weighting exists for a 5/11 code (and also for a 6/11 code, by an earlier argument). In this way, codes such as the following are shown to be unacceptable for digit weighting:

- $5/n$, when n has the form $8k+1$ or $8k+3$
- $6/n$, when n has the form $8k+1$, $8k+2$, $8k+3$, or $8k+5$
- $7/n$, when n has the form $8k+1$, $8k+3$, or $8k+5$
- $8/n$, when n has the form $16k+1$, $16k+3$, $16k+5$, or $16k+7$

etc.

Appendix III

Incomplete Decoding Trees

When N is a power of two ($N=2^{n_b}$), it is known[3] that the minimal cost of the complete conventional decoding tree in terms of AND-gate inputs is

$$c_b(N) = 2N + c_b(2^{r_b}) + c_b(N/2^{r_b})$$

where $r_b = [n_b/2]$. The structure of the n_b-tree is as follows. Each of the N outputs is generated by a two-input AND-gate (the first term), one of whose inputs is provided by one of the outputs of a complete r_b-tree having 2^{r_b} outputs (the second term), and the other by one of the outputs of an $(n_b\text{-}r_b)$-tree having $2^{n_b-r_b}$ outputs (the third term). Each of these two smaller trees is realized by exactly the same dichotomization as was applied to the original tree. The dichotomization is continued until only 1-trees and 2-trees remain. For these, the minimal costs can be determined easily by inspection

$$c_b(2) = 0. \qquad c_b(4) = 8.$$

As N is reduced below 2^{n_b}, a reduction from the cost of the complete tree can be effected in two ways. Not only can the gates corresponding to the now undesired higher-numbered outputs be eliminated, but it may be possible, as a result, to remove parts of the smaller trees, in the event that not all of their own outputs are needed. If the complete tree is "pruned" in this fashion, the above formula for $c_b(N)$ continues to hold approximately as N is reduced below 2^{n_b}.

Actually, a few additional gate-inputs can be saved as a result of certain degeneracies for small n_b, and for special values of N such as $2^{n_b-1}+1$. Examination of simple cases reveals that $c_b(2)=0$, $c_b(3)=2$. For $N \geq 4$, the cost of an incomplete tree pruned from a complete one in the manner described is given by

$$c_b(N) = 2N + c_b(2^{r_b}) + c_b\left(1 + \left[\frac{N-1}{2^{r_b}}\right]\right),$$

except that if the fractional part discarded by the bracket operation happens to be zero, $c_b(N)$ is given by the lesser of this expression and

$$c_b(N) = c_b(N-1) + 1 \qquad \text{if} \quad \mu = 1,$$

or

$$c_b(N) = c_b(N-1) + 2 + \mu \quad \text{if} \quad \mu > 1,$$

where μ = the number of 1's in the binary representation of the number $N-1$; *i.e.*,

$$\mu = (N-1) - \sum_{v=1}^{\infty}\left[\frac{N-1}{2^v}\right];$$

e.g., if $N=9$, the binary form of $N-1$ is 1000, so $\mu=1$. $c_b(9)$ is then the lesser of $2\cdot 9 + c_b(4) + c_b(3) = 28$ and $c_b(8)+1=25$; namely, $c_b(9)=25$.

The above expression for $c_b(N)$ appears to be minimal for the class of incomplete trees derived by pruning minimal complete trees. Whether some less costly incomplete trees can be found by another method, such as pruning complete trees that are initially more costly than the minimal form, is not known.

It may also be possible to evolve less costly binary ring-type counters by modifying the counter itself to skip states in the binary counting sequence, instead of counting systematically from 0 to $N-1$. If this were done properly, the saving in the cost of the decoding tree might exceed the extra gatery required to modify the basic counting pattern. This possibility has not been investigated.

Acknowledgment

The author wishes to acknowledge the assistance of S. Feferman, B. Elspas, and J. Goldberg, who participated in various ways in this study and its written presentation.

Determination of the Irredundant Normal Forms of a Truth Function by Iterated Consensus of the Prime Implicants*

THOMAS H. MOTT, JR.†

Summary—This paper describes a new algebraic way of determining irredundant forms from the prime implicants. The method does not require using the developed normal form, and it makes novel application of Quine's technique of iterative consensus-taking. Thus, by applying repeatedly the rule of consensus to the prime implicants, it is possible to derive a list of implication relations that express the necessary and sufficient conditions of eliminability of the prime implicants in terms of which the irredundant normal forms can be computed. The extension of Quine's technique to this phase of simplification serves to shorten considerably the logical machinery needed for complete solution of the simplification problem. By the same token, it renders the method suitable for use with a digital computer.

I. Introduction

SINCE C. E. Shannon's correlation of the formulas of truth-function logic with switching circuits, a major problem of switching algebra has been to devise a general procedure of obtaining simplest equivalents of a formula. The general problem has, so far, proved refractory. However, an interesting specialization of the problem as studied by W. V. Quine and others is to find the set of simplest normal equivalents.

Quine [13] showed that such forms are disjunctions of certain fundamental formulas called *prime implicants*, and he later described a way of finding all the prime implicants of a formula using a technique known as *iterative consensus-taking* [14]. For the simplification problem then, it remained to develop a suitable algorithm for moving from the disjunction of all the prime implicants of a formula to its set of simplest normal forms. This requires finding a way of selecting from the list of prime implicants shortest subsets of prime implicants whose disjunctions are equivalent to the formula.

A number of different solutions to the problem have appeared. Depending upon the representation used to achieve simplification, they may be classified as chart [1]–[3], geometric [4]–[6], or algebraic methods [7]–[14]. Geometric methods associate with a truth function in n letters its cell complex in the n cube, and operate upon the latter to obtain simplest forms. Chart and algebraic methods, on the other hand, resort to transformations that are purely truth-functional in character. Algebraic methods further vary according to whether the method requires preliminary expansion of a formula into developed normal form or not. Such expansion is necessary, for example, in the methods of McCluskey [8], and Petrick [12], whereas the table of ratio functions by Gazale [7] determines irredundant forms directly from the list of prime implicants themselves.

In the present paper we describe a new algebraic way of obtaining simplest forms from the prime implicants. The method does not require use of the cumbersome developed normal form, and makes novel use of Quine's technique of iterative consensus-taking. The extension of Quine's technique to this phase of simplification shortens considerably the logical machinery needed for complete solution of the simplification problem. By the same token, it renders the method suitable for use with a digital computer.

Section II will serve to introduce the reader to the terminology used in the paper. Section III introduces the method and contains theorems which establish its validity. Section IV contains an informal explanation of Section III and the solution and discussion of a sample problem. Section V, finally, discusses a more complex example and contains some informal suggestions as to an optimal programming procedure to follow for machine use of the method in complicated cases.

II. Preliminary Definitions

We begin by defining some logical terms common in the literature. A *literal* is either a letter or negation of a letter (represented by a letter with a bar over it). A *fundamental formula* (FF) is either a literal or a conjunction of literals containing no single letter twice. Thus $\bar{p}$ and pqr are FF's, but pqp, $\bar{p}q\bar{p}$ and $pq\bar{p}$ are not FF's. A *normal formula* (or disjunctive normal formula) is either a FF or a disjunction of FF's. Thus $\bar{p}$ and $pq \vee \bar{p}rs$ are normal formulas ("$\vee$" is the sign for disjunction) but $pq \vee \bar{p}rp$ is not, since normal formulas cannot contain conjunctions with a repetitious letter. We say that a normal formula is in *developed* form if every FF of the formula contains every letter that the formula itself contains. Thus $\bar{p}q$ and $pq\bar{r} \vee \bar{p}\bar{q}\bar{r}$ are normal formulas in developed form. $\bar{p}q \vee qr \vee \bar{p}q\bar{r}$ is a normal formula which is not developed.

A normal formula will be *irredundant* if it contains no FF that can be dropped, nor any occurrences of literals that can be dropped, without breach of equivalence. Irredundant normal formulas therefore contain no superfluous FF and none of the FF's contain superfluous literals.

* Received by the PGEC, January 18, 1960; revised manuscript received, February 25, 1960. Portions of this paper were presented to the Symposium on Switching Algebra, held in conjunction with the International Conf. on Information Processing, Paris, France, June, 1959. The research for this paper was supported in part under contract with the U. S. Bureau of Ships, Contract No. NObrs 72728, while the author was a member of the Mathematical Research Dept., Remington Rand Univac, St. Paul, Minn. Thanks are due to former colleagues, especially A. A. Cohen and R. A. Zemlin, who gave much helpful criticism.

† RCA Laboratories, Princeton, N. J.

If a formula comes out true under all assignments of truth values to the letters, we say the formula is *valid* (or tautologous); if it comes out true under some and false under others, we say the formula is *consistent*. One formula *implies* another if there is no assignment of truth values which makes the first formula true and the second false. Two formulas are *equivalent* if each implies the other.

We now define the key concept of the simplification problem, that of prime implicant, and two auxiliary notions whose importance for the simplification project Quine established once and for all [14]. An FF will be called a *prime implicant* of a normal formula Φ if it implies Φ and implies no shorter FF which implies Φ. If ϕ and ψ are FF's such that there is exactly one literal which occurs affirmed in ϕ and negated in ψ, or vice versa, the conjunction $\phi\psi$, with the opposing literals and any duplicate literal deleted, will be called the *consensus* of ϕ and ψ. Thus the consensus of $p\bar{r}\bar{s}$ and $q\bar{r}s$ is $pq\bar{r}$, and the consensus of $\bar{p}$ and pqs is qs. One FF will be said to *subsume* another if all the literals whose conjunction is the latter FF are among the literal whose conjunction is the former. Thus $p\bar{q}rs$ subsumes $\bar{q}s$ but does not subsume pqr or $\bar{q}rt$.

By an *implication* or *implication relation*, we shall mean any compound formula expressed as $\phi \rightarrow \Psi$ where ϕ and Ψ denote, respectively, the antecedent (or implicant) and consequent of the relation. An implication will be valid if its antecedent implies its consequent. For purposes of this paper, we shall always confine attention to valid implications whose antecedents are FF's and whose consequents are normal formulas. One implication will be said to *subsume* another if both have the same antecedent and all the FF's whose disjunction is the consequent of the latter implication are among the FF's whose disjunction is the consequent of the former. Thus $\phi_5 \rightarrow \phi_2 \vee \phi_3 \vee \phi_6$ subsumes $\phi_5 \rightarrow \phi_3 \vee \phi_6$. Finally, it will be necessary to refer to *irredundant* implications, and we shall mean by this any valid implication whose consequent is irredundant.

III. The Dual Role of Iterative Consensus-Taking

Although the goal of simplification is to secure simplest normal forms, the emphasis is rather upon finding a procedure for determining irredundant normal forms. This is because "simplest" normal forms are viewed as special instances of the irredundant forms. Which among the latter actually constitute the simplest forms of a formula therefore depends upon the criterion of simplicity that is used. Since a number of reasonable definitions of simplicity may be chosen, it is useful to limit the simplification problem to that of finding the irredundant equivalents of any normal formula.

This fact serves to explain the interest in prime implicants. Recall from the preceding section that prime implicants are necessarily shortest FF's which imply a formula. They therefore contain no superfluous literals. An obvious way of determining the irredundant equivalents of a formula is, then, to begin with the disjunction of all the prime implicants and to weed out superfluous FF's by locating and removing all largest combinations of simultaneously dispensable prime implicants. This will result in disjunctions of prime implicants having the property that none of the prime implicants, and therefore, no occurrences of literals therein, can be removed from any of the disjunctions without sacrificing equivalence. When all such combinations have been removed, we have the set of irredundant normal equivalents.

The starting point of the simplification project is, then, to elicit all the prime implicants. Quine [14] explained a speedy and direct method that converts a normal formula into the disjunction of all its prime implicants simply by continued use of the two following rules:

1) delete subsuming FF's,

2) adjoin as an additional disjunct the consensus of FF's.

Quine's method does not require expansion of the formula into developed normal form. Rather, it arrives at prime implicants by applying the above rules directly to the formula as given in normal form. When the rules cease to be applicable further, we have the disjunction of all the prime implicants.

For the second (and final) phase of simplification, the problem is to move from the disjunction of all the prime implicants to the irredundant normal equivalents. Since the latter, we saw, correspond to disjunctions of prime implicants, none of which are superfluous, the problem is actually the reduction of the original redundant disjunction containing all the prime implicants to the set of equivalent irredundant disjunctions. The main contribution of the present paper lies in the extension of Quine's rules to this phase of the simplification problem. Specifically, we intend to show that rule 2) above, when iteratively applied to the disjunction of all the prime implicants, results in a set of consensus FF's from which a list of implications can be deduced. Then, if *subsuming* implications are deleted from the list, the remaining relations express the necessary and sufficient conditions for eliminating each prime implicant. Once these relations are known, it is a simple matter to arrive at the shortest irredundant disjunctions.

Our starting point, then, is the notion of an eliminable prime implicant. In view of the validity of the following equivalence,

$$(\phi \vee \Psi \equiv \Psi) \equiv (\phi \rightarrow \Psi),$$

an FF is eliminative with respect to the remaining FF's of a normal formula if and only if it implies the disjunction of the remaining FF's. It may happen that a prime implicant fails to imply any disjunction of remaining prime implicants. Following Quine, we call such FF's *core* prime implicants or core FF's. The above equivalence serves to establish that a disjunction of prime implicants is irredundant if none of the prime implicants in the disjunction implies the remainder of the disjunction. It is obvious that if a formula has core

rime implicants, the latter must appear in every ir-edundant equivalent of the formula. It is further ob-ious that if an eliminable prime implicant appears in n irredundant solution, none of the subsets of other rime implicants whose disjunctions are implied by it an appear in the same solution. On the other hand, if a rime implicant is absent from an irredundant form, ome subset of prime implicants which is implied by it s necessarily present. We express this by saying that for very prime implicant of a formula, either the prime mplicant itself or one of the subsets of other prime mplicants whose disjunctions it implies (if it is elimina-ive) appears in every irredundant solution of the ormula.

Gazale [7] has shown an interesting way to compute he irredundant forms, given these facts. The technique s one which we shall follow in this paper and is related o one that Petrick [12] has used on occasion. It consists f converting a formula from conjunctive form to dis-unctive normal form. Let each prime implicant be epresented by a capital letter and each subset of prime mplicants whose disjunction is implied by a prime im-licant by a conjunction of capital letters. Then a ogical formula in conjunctive form is written, one con-unct for each prime implicant expressing the disjunc-ion of only the letter and conjunctions of letters for hat prime implicant and the subsets (if any) of other rime implicants which it implies. When this formula s converted into disjunctive normal form, each dis-unct that does not subsume another represents an ir-edundant solution.

We now show that the subsets of remaining prime im-licants which the eliminable prime implicants imply re obtainable directly from the list of prime implicants y continued use of Quine's rule of consensus. This will e proved as Theorem 3 below. The two intervening heorems, which are needed as lemmas, establish some ecessary properties that every consensus FF has in ommon.

Theorem 1): If γ is the consensus of α and β, then $\gamma \rightarrow \alpha \vee \beta$.

Proof: All the letters of γ appear in the disjunction f α and β. α and β each contain at least one letter, and t most one, not common to γ, which together take the orm of p and $\bar{p}$. An assignment of truth values which nakes γ true also makes the disjunction $\alpha \vee \beta$ true, since he latter upon such assignment reduces to $p \vee \bar{p}$.

Theorem 2): If γ is the consensus of α and β, and α is he consensus of δ and λ, then $\gamma \rightarrow \beta \vee \delta \vee \lambda$.

Proof: By Theorem 1, $\alpha \rightarrow \delta \vee \lambda$ and $\gamma \rightarrow \alpha \vee \beta$. By *modus ponens*, if γ is true, then the disjunction of α and is true. If α is true, then again by *modus ponens* the isjunction of δ and λ is true, and if so, the disjunction f β, δ and λ is true. If β is true, the disjunction of β, δ nd λ is true. Hence, if γ is true, the disjunction of β, and λ is true.

It will be noticed here that Theorem 2 and variants f the theorem allow for the possibility of implications n which the prime implicants appearing in the conse-equent are greater than two in number. The theorem further brings out the relevance to our goal of obtaining such relations of listing each consensus FF and the pair of FF's which give rise to it as the process of consensus-taking continues. When the operation is applicable no further (*i.e.*, no new consensus results), and after apply-ing Theorems 1 and 2, then, as will be proved next, we have all the subsets of remaining prime implicants which the eliminable prime implicants imply.

Theorem 3): Given any consistent formula Φ whose set of prime implicants consists of FF's $\phi_1, \phi_2, \cdots, \phi_n$ and, given further any subset of prime implicants there-of whose disjunction we denote by Ψ such that ϕ_i is not a disjunct of Ψ, then $\phi_i \rightarrow \Psi$ if and only if ϕ_i is among the set of consensus FF's that result from adjoining to Ψ all FF's that are consensus of pairs of disjuncts.

Proof: Sufficiency condition—by Theorem 1 and vari-ants of Theorem 2. Necessity condition—ϕ_i implies Ψ. ϕ_i subsumes no shorter FF ψ such that ψ implies Ψ; other-wise ψ must imply Φ, which contradicts the hypothesis that ϕ_i is a prime implicant of Φ. Hence ϕ_i is a prime implicant of Ψ. The remainder of the proof is due to Quine [14] and shows that, as long as ϕ_i is not a dis-junct of Ψ, Ψ is susceptible to having additional FF's adjoined to it in accordance with the following equiv-alences:

$$\alpha \vee \bar{\alpha}\phi \equiv \alpha \vee \bar{\alpha}\phi \vee \phi,$$

$$\bar{\alpha} \vee \alpha\phi \equiv \bar{\alpha} \vee \alpha\phi \vee \phi,$$

$$\alpha\phi \vee \bar{\alpha}\psi \equiv \alpha\phi \vee \bar{\alpha}\psi \vee \phi\psi.$$

ϕ_i contains only letters of Ψ and subsumes no FF of Ψ. Let χ be a longest FF such that χ subsumes ϕ_i, sub-sumes no FF of Ψ, and contains only letters of Ψ. χ will lack some letter α of Ψ. Hence each of the longer FF's $\alpha\chi$ and $\bar{\alpha}\chi$ subsumes an FF of Ψ, since they subsume ϕ_i and contain only letters of Ψ. Moreover, the FF of Ψ subsumed by $\alpha\chi$ and $\bar{\alpha}\chi$ must contain α and $\bar{\alpha}$, re-spectively, since they were not subsumed by χ alone. But they cannot be simply α and $\bar{\alpha}$. Hence, there are only three possible cases. They are either: 1) $\alpha\psi$ and $\bar{\alpha}$, where χ subsumes ψ ; or 2) α and $\bar{\alpha}\psi'$, where χ subsumes ψ'; or 3) $\alpha\psi$ and $\bar{\alpha}\psi'$. In either case, Ψ is susceptible to having an additional FF adjoined to it in accordance with the above equivalences, since ψ, ψ', and $\psi\psi'$ (with duplicate literals deleted) are readily seen to meet the requirement of a consensus FF: they contain no letter both affirmed and negated, since each is subsumed by χ.

We remarked earlier that a prime implicant need not imply the disjunction of the remaining prime implicants. We called such prime implicants core FF's, and the core itself is simply the disjunction of all core FF's. The following theorem establishes that the core of Φ, if it exists, is the disjunction of those prime implicants and only those which are not among the set of FF's that derive from the prime implicants through continued ap-plication of the rule of consensus.

Theorem 4): Given any consistent formula Φ whose set of prime implicants consists of FF's $\phi_1, \phi_2, \cdots, \phi_n$, then

1) ϕ_i is a core FF of Φ if and only if ϕ_i is not a prime implicant of $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \vee \cdots \vee \phi_n$,

2) ϕ_i is not a prime implicant of $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \vee \cdots \vee \phi_n$ if and only if ϕ_i is not among the set of FF's that result from adjoining to the prime implicants of Φ, in disjunctive form, all FF's that are consensus of pairs of disjuncts.

Proof 1): By the definitions of prime implicant and core FF. ϕ_i is not an implicant of $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \vee \cdots \vee \phi_n$ unless it is a prime implicant of it, since $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \vee \cdots \vee \phi_n$ implies Φ, and ϕ_i is a prime implicant of Φ.

Proof 2): By the contrapositive of Theorem 3 for the case where $\Psi = \phi_1 \vee \phi_2 \vee \cdots \vee \phi_{i-1} \vee \phi_{i+1} \vee \cdots \vee \phi_n$. As pointed out in the proof of Theorem 3, if ϕ_i is an implicant of Ψ, it is a prime implicant of Ψ.

It follows from Theorem 4 that the prime implicants which do not appear in any antecedent of the set of implications that result from applying Quine's rule of consensus to the prime implicants are core FF's, and so appear in every irredundant normal form of the expression. The complete set of such implications involves redundancy. This is because it may happen that a given prime implicant ϕ_i is the consensus of two FF's such that one of them, in turn, is the consensus of ϕ_i and a third FF. Consequently, the same prime implicant may appear as both the antecedent of an implication and as a disjunct within the consequent. Such occurrences will be called *trivially redundant* implications. (The relation $\phi_5 \rightarrow \phi_2 \vee \phi_3 \vee \phi_5$, for example, is trivially redundant.) It may further happen that among the prime implicants whose disjunction is implied by a given prime implicant ϕ_i, there appears a subset of prime implicants, whose disjunction ϕ_i also implies. Relations of this kind are what we called in Section II *subsuming* implications. By virtue of the following logical principle,

$$(\phi \rightarrow \Psi) \rightarrow [(\phi \rightarrow \Psi \vee \psi \vee \cdots) \equiv (\phi \rightarrow \Psi)],$$

one is always justified in dropping subsuming implications from the list of all such relations. Hence if one also drops all occurrences of trivially redundant relations, the remaining relations and only those constitute the set of irredundant implications. We therefore put forward the following method.

Method

Given the set of FF's that derive from the prime implicants of Φ by applying the rule of consensus in all possible ways, and given further the list of implications that follow therefrom, subject to the following restrictions, a) drop trivially redundant implications and b) drop subsuming implications, then,

1) remaining entries of the list constitute all the irredundant implications which hold for the eliminable prime implicants of Φ, and

2) any prime implicant not appearing as an antecedent of any of the above implications is a core prime implicant of Φ.

IV. A Sample Problem

To obtain the irredundant forms from the list of prime implicants, we divide the work into three stages. The first stage is to elicit the set of consensus FF's by applying the rule of consensus to the prime implicants and noting the pairs of FF's which have a consensus. The second stage is to derive, from the set of consensus FF's thus obtained, the list of irredundant implications. The third stage is to compute the irredundant forms on the basis of the derived implications. We have some suggestions as to a good procedure for the first stage. They will be presented in this section, along with the complete solution of a sample problem. We have no theoretically satisfactory procedure for the second stage. In practice, the business of deducing the irredundant implications from the consensus FF's proceeds by exhaustion: following all possible paths that lead out from a given consensus FF and comparing all the resulting implications for irredundancy. *Since all the information that is needed to compute the irredundant forms is already present at the conclusion of the first stage, it would be desirable to find a quicker means of moving to the irredundant forms than by way of the irredundant implications.* However, we have not seen a way to manage this. Finally, we consider some suggestions common in the literature for speeding up the third stage, which will be illustrated by a more complicated example in Section V.

Recall from Section III that the consensus operation requires iterating until no new FF's are obtained. This is true whether the operation is used to convert a formula into the disjunction of prime implicants or to derive the irredundant implications from the prime implicants. In practice, adherence to this requirement results in a laborious and often unwieldy process of consensus-taking. To eliminate this difficulty, it is possible to shorten the process in the manner described below. Although the shortened procedure will not test for consensus among all possible pairs of FF's, nevertheless, it can be depended upon to yield all consensus FF's which do not subsume another FF. This means that it will always yield all the prime implicants and, further, that it will not omit any consensus FF's which give rise to an irredundant implication. (We omit the proof of these statements.) The procedure is thorough, reasonably short and sufficiently systematic to be programmed on a computer.

We describe the procedure here only as it applies to the second phase of simplification. Given the list of prime implicants, each item on the list is tested for consensus with all succeeding items. If there results a consensus which is not a prime implicant, the new FF is added to the end of the list. Testing is continued in this manner until every item on the list, including added ones, has been tested for consensus with all succeeding items and none remains to be tested. The procedure can be represented in tabular form by letting each prime implicant (and each new FF) correspond to a row and column of a table and letting the initial column contain the same number of rows as there are columns, and

ll succeeding columns contain one less row than the olumn immediately preceding it (see Table I below). 'o test for consensus, it is necessary to check an FF only gainst those FF's whose rows intersect the column for hat FF. Should a consensus result, it is entered at the ntersection of the column with the appropriate row. Vhenever a consensus appears which is not a prime mplicant, an extra row and column must be added to he table corresponding to the new FF. When every olumn has been checked against its rows, the result is a able of consensus FF's which forms the basis for com-uting the irredundant forms.

The following sample problem, whose irredundant olutions we shall find in this section, will serve to intro-uce the table. Given a truth function Φ whose dis-unction of prime implicants $\phi_1, \phi_2, \cdots, \phi_6$ is as follows:

$$\begin{array}{cccccc} \bar{p}\bar{q} \vee \bar{p}r \vee qr \vee p\bar{q} \vee p\bar{r} \vee \bar{q}\bar{r} \\ A \quad B \quad C \quad D \quad E \quad F \end{array} \tag{1}$$

he first step is to derive the table of consensus FF's by pplying the rule of consensus to (1) in the manner just escribed. Each ϕ_i is tested for consensus only with hose ϕ_j's whose rows intersect the column correspond-ng to ϕ_i. When every column has been so tested and the esulting consensus FF's entered at the appropriate in-ersections of rows and columns, we have the table of onsensus FF's as shown below. Since every consensus F which appears in the table is a prime implicant, the able contains only rows and columns corresponding to rime implicants.

TABLE I

CONSENSUS FF'S FOR $\bar{p}\bar{q} \vee \bar{p}r \vee qr \vee p\bar{q} \vee p\bar{r} \vee \bar{q}\bar{r}$

		A $\bar{p}\bar{q}$	B $\bar{p}r$	C qr	D $p\bar{q}$	E $p\bar{r}$	F $\bar{q}\bar{r}$
A	$\bar{p}\bar{q}$	—	$\bar{p}r$				
B	$\bar{p}r$		—	qr			
C	qr	$\bar{p}\bar{r}$ B		—	pq		
D	pq		qr C		—	$p\bar{r}$	
E	$p\bar{r}$	$\bar{q}\bar{r}$ F		pq D		—	$\bar{q}\bar{r}$
F	$\bar{q}\bar{r}$		$\bar{p}\bar{q}$ A		$p\bar{r}$ E		—

This completes the first stage of the problem. The econd stage is to determine the list of irredundant im-lications starting with the basic implications which able I yields:

$$\begin{array}{lll} (B) & \bar{p}r \rightarrow \bar{p}\bar{q} \vee qr & (A\ C), \\ (F) & \bar{q}\bar{r} \rightarrow \bar{p}\bar{q} \vee p\bar{r} & (A\ E), \\ (C) & qr \rightarrow \bar{p}r \vee pq & (B\ D), \\ (A) & \bar{p}\bar{q} \rightarrow \bar{p}r \vee \bar{q}\bar{r} & (B\ F), \\ (D) & pq \rightarrow qr \vee p\bar{r} & (C\ E), \\ (E) & p\bar{r} \rightarrow pq \vee \bar{q}\bar{r} & (D\ F). \end{array} \tag{2}$$

It is easily verified that any new implication which logically derives from any of the above is either trivially redundant or else subsumes one of them. Consequently, (2) represents the complete list of irredundant implications. This completes the second stage of the problem.

For the third stage—computing the irredundant forms on the basis of the above implications—we employ the technique of Gazale [7] mentioned earlier. Letting each prime implicant be represented by a capital letter as shown in (1) and letting a conjunction of capital letters represent each subset of prime implicants whose disjunction is implied by a prime implicant as indicated in (2), we write a formula in conjunctive form, one conjunct for each prime implicant, as follows:

$$(A \vee BF)(B \vee AC)(C \vee BD)$$
$$(D \vee CE)(E \vee DF)(F \vee AE) \tag{3}$$

where each conjunct expresses the disjunction only of the letter and conjunction(s) of letters for that prime implicant and the subset(s) of other prime implicants whose disjunction(s) it implies. Application of the distributive law will then yield a formula in disjunctive normal form such that every disjunct that does not subsume another represents an irredundant solution. Hence "multiplying out" (3) and deleting subsuming disjuncts yields

$$ACE \vee BDF \vee ABDE \vee ACDF \vee BCEF. \tag{4}$$

The five irredundant normal forms are therefore

$$\begin{array}{ll} \bar{p}\bar{q} \vee qr \vee p\bar{r} & \text{corresponding to } ACE, \\ \bar{p}r \vee pq \vee \bar{q}\bar{r} & \text{corresponding to } BDF, \\ \bar{p}\bar{q} \vee \bar{p}r \vee pq \vee p\bar{r} & \text{corresponding to } ABDE, \\ \bar{p}\bar{q} \vee qr \vee pq \vee \bar{q}\bar{r} & \text{corresponding to } ACDF, \\ \bar{p}r \vee qr \vee p\bar{r} \vee \bar{q}\bar{r} & \text{corresponding to } BCEF. \end{array}$$

V. Some Programming Heuristics

In this section, we shall offer some heuristic considerations which will aid in applying the method in complicated cases. The second stage of the method calls for finding all shortest disjunctions of prime implicants which the eliminable prime implicants imply. Although the logical operation required for this stage is simple enough, in many cases the process of carrying it out may often become unwieldy. What is required is to test for irredundancy among implications which are not included among those yielded by the consensus table, but which nonetheless follow from them as logical consequences. This means that in a sense, the information needed to compute irredundant forms is present in the table of consensus at the conclusion of the first stage; therefore, to that extent the second stage is superfluous. Hence it would be desirable to bypass it and use only

the set of implications yielded directly by the consensus table to compute the minimal forms. So far, we have not seen how to do this. Meanwhile, to aid the logical task of the second stage in complicated cases, we introduce the use of "implication trees" and will illustrate them later in our discussion of a more complex example.[1]

It is sometimes possible to speed up the operation of the third stage of the method through the introduction of one or more of the following simplification procedures. Recall that for the third stage we write an auxiliary formula in conjunctive form in which each conjunct expresses the necessary and sufficient conditions for the appearance of that prime implicant in an irredundant solution.

1) If the table of consensus reveals the presence of core FF's, replace all occurrences of capital letters corresponding to these FF's in the auxiliary formula by "1." Since core FF's must appear in every irredundant solution, they can always be added later to the results that obtain when we omit them from consideration.

2) If the table of consensus reveals a condition in which a prime implicant implies the disjunction of core FF's, replace all occurrences of the capital letter corresponding to it in the auxiliary formula by "0." This is because prime implicants of this sort cannot appear in any irredundant solution.

3) Delete any conjunct which subsumes another as a result of applying operations 1) and 2) to the auxiliary formula.

4) In multiplying out the auxiliary formula, delete subsuming disjuncts as the multiplication proceeds. This is because deletion of subsuming disjuncts from partial results produces exactly the same outcome as deletion after complete multiplication, but reduces the number of disjuncts that must be deleted.

Let us now see how the use of implication trees and these four rules aid in applying the method in a complicated case. Given a truth function Φ whose disjunction of prime implicants is as follows:

$$\begin{array}{cccccccc} \bar{d}e \vee cd\bar{e} \vee \bar{a}cd \vee \bar{a}ce \vee a\bar{b}d \vee a\bar{b}e \vee \bar{b}cd \vee \bar{b}ce \\ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \end{array} \tag{5}$$

we wish to find the irredundant forms of Φ. First, the table of consensus FF's is determined. Since there result two consensus FF's which are not prime implicants, it is necessary that the table have two extra rows and columns corresponding to the new FF's. They are $a\bar{b}ca$ represented by X in Table II, and $\bar{b}cde$, represented by Y.

[1] For an excellent treatment of a problem similar in nature to that of the second stage of our method, see paper by A. L. Leiner, *et al.*, "Organizing a network of computers to meet deadlines," *Proc. EJCC*, pp. 115–128; December, 1957, where matrix techniques are introduced to determine from an incomplete list of explicit priority requirements all the implicit priority statements.

TABLE II

CONSENSUS FF'S FOR (5)

		A $\bar{d}e$	B $cd\bar{e}$	C $\bar{a}cd$	D $\bar{a}ce$	E $a\bar{b}d$	F $a\bar{b}e$	G $\bar{b}cd$	H $\bar{b}ce$	X $a\bar{b}cd$	Y $\bar{b}cde$
A	$\bar{d}e$	—	$cd\bar{e}$								
B	$cd\bar{e}$		—	$\bar{a}cd$							
C	$\bar{a}cd$	$\bar{a}ce$ D		—	$\bar{a}ce$						
D	$\bar{a}ce$		$\bar{a}cd$ C		—	$a\bar{b}d$					
E	$a\bar{b}d$	$a\bar{b}e$ F		$\bar{b}cd$ G	$\bar{b}cde$ Y	—	$a\bar{b}e$				
F	$a\bar{b}e$		$a\bar{b}cd$ X	$\bar{b}cde$ Y	$\bar{b}ce$ H		—	$\bar{b}cd$			
G	$\bar{b}cd$	$\bar{b}ce$ H						—	$\bar{b}ce$		
H	$\bar{b}ce$		$\bar{b}cd$ G						—	$a\bar{b}cd$	
X	$a\bar{b}cd$			$\bar{b}cd$ G	$\bar{b}cde$ Y					—	$\bar{b}cde$
Y	$\bar{b}cde$										—

From Table II there results the following set of primitive implications having the property that any irredundant implication which fails to be included is a logical consequence of the set:

$$\begin{aligned} &D \to A \vee C \qquad && G \to C \vee E, \\ &F \to A \vee E \qquad && Y \to C \vee F, \\ &H \to A \vee G \qquad && G \to C \vee X, \\ &C \to B \vee D \qquad && Y \to D \vee E, \\ &X \to B \vee F \qquad && H \to D \vee F, \\ &G \to B \vee H \qquad && Y \to D \vee X. \end{aligned} \tag{6}$$

(For convenience, capital letters are written in place of the actual prime implicants. The reader is therefore reminded that this usage in which capital letters are treated as abbreviations for the prime implicants is altogether distinct from that of the third stage in which capital letters denote the presence of prime implicants in irredundant solutions.)

In order to formulate for each prime implicant the necessary and sufficient conditions for its occurrence in a minimal solution, it is necessary to establish those irredundant implications which, though missing from (6), are implied therein. Since A, B and E do not occur as antecedents in (6), they are core FF's and therefore appear in every minimal solution. F implies $A \vee E$ and so implies the core, hence F is absolutely eliminable. This leaves C, D, G and H. It is easy to verify that (6) does not give rise to any new irredundant implication whose antecedent is C or D. With G and H, however, the situation is more complicated. With the aid of the implication tree shown in Fig. 1, however, it is relatively easy to determine the irredundant implications for G and H.

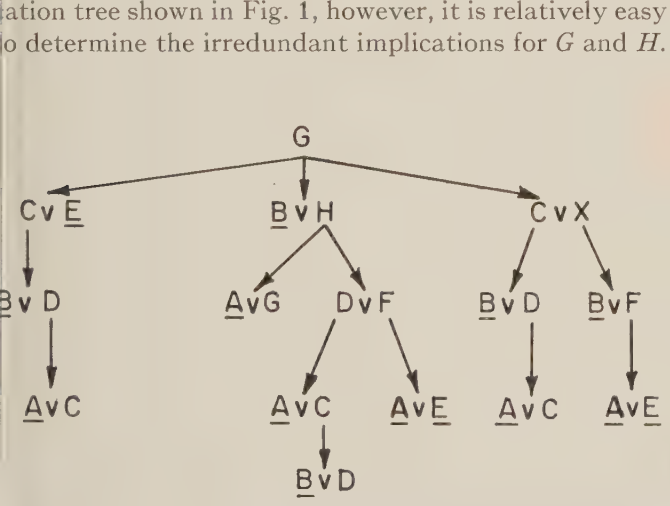


Fig. 1—Implication tree for deriving irredundant implications missing from (6). The figure is based on Theorem 2 and variants of the theorem. Thus if $G \to C \vee X$ and $X \to B \vee F$, then $G \to B \vee C \vee F$, etc.

The tree corresponds to a graphic representation of (6), and paths in it terminate when a prime implicant is reached that is either a core FF (underlined in the figure) or whose occurrence marks its second appearance in the path.

From Fig. 1 we can now derive the complete list of irredundant implications for G and H as follows:

$$\begin{aligned} &G \to B \vee H \qquad && H \to A \vee G, \\ &G \to C \vee E \qquad && H \to D \vee F, \\ &G \to B \vee C \vee F \qquad && H \to A \vee C \vee E, \\ &G \to B \vee D \vee E \qquad && H \to A \vee C \vee F, \\ &G \to B \vee D \vee F \qquad && H \to A \vee D \vee E. \end{aligned} \tag{7}$$

This completes the second stage of the problem. The third stage is to write the auxiliary formula expressing the necessary and sufficient conditions for the occurrences of prime implicants in minimal solutions, as determined by (6) and (7):

$$\begin{aligned} &(A)(B)(C \vee BD)(D \vee AC)(E)(F \vee AE) \\ &\qquad \cdot (G \vee BH \vee CE \vee BCF \vee BDE \vee BDF) \\ &\qquad \cdot (H \vee AG \vee DF \vee ACE \vee ACF \vee ADE). \end{aligned} \tag{8}$$

By virtue of rule 1), all occurrences of A, B and E can be omitted from (8) and introduced later, since these letters represent core FF's. Further, since F represents a prime implicant which implies the core, rule 2) permits $F \vee AE$ to be omitted as well as all other occurrences of F in the formula; (8) thus reduces to

$$(C \vee D)(G \vee H \vee C \vee D), \tag{9}$$

which in accordance with rule 3) further reduces to

$$C \vee D. \tag{10}$$

The two irredundant normal forms of Φ are therefore $\bar{d}e \vee cd\bar{e} \vee \bar{a}cd \vee a\bar{b}d$, corresponding to $ABCE$, and $\bar{d}e \vee cd\bar{e} \vee \bar{a}ce \vee a\bar{b}d$, corresponding to $ABDE$.

Acknowledgment

The author is indebted to Professor P. Rosenbloom for reading and suggesting several changes in an early draft of the paper and Professor N. Scott for the opportunity of presenting the material in two lectures at the recent summer session of the College of Engineering, University of Michigan, Ann Arbor. Lastly, thanks are due to R. O. Winder of the RCA Laboratories, Princeton, N. J., for his help and interesting suggestions.

Bibliography

Chart Methods

[1] Harvard Computation Lab., "Annals of the Computation Laboratory," in "Synthesis of Electronic Computing and Control Circuits," Harvard University Press, Cambridge, Mass., vol. 27; 1951.

[2] M. Karnaugh, "The map method for synthesis of combinational logic circuits," *Commun. and Electronics, Trans. AIEE*, pt. I, vol. 72.

[3] E. W. Veitch, "Third and Higher Order Mimimal Solutions of Logical Equations," presented at Symp. on Switching Algebra, ICIP, Paris, France; June, 1959.

Geometric Methods

[4] B. Harris, "An algorithm for determining minimal representations of a logic function," IRE Trans. on Electronic Computers, vol. EC-6, pp. 103–108; June, 1957.
[5] R. H. Urbano and R. K. Mueller, "A topological method for the determination of the minimal forms of a Boolean function," IRE Trans. on Electronic Computers, vol. EC-5, pp. 126–132; September, 1956.
[6] J. P. Roth, "Combinatorial Topological Methods in the Synthesis of Switching Circuits," *Proc. Internat. Symp. on the Theory of Switching*, Harvard University, Cambridge, Mass.; April, 1957.

Algebraic Methods

[7] M. J. Gazale, "Irredundant disjunctive and conjunctive forms of a Boolean function," *IBM J. Res. & Dev.*, vol. 1, pp. 171–176; April, 1957.
[8] E. M. McCluskey, "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.
[9] T. H. Mott, "An algorithm for Determining Minimal Forms of an Incomplete Truth Function," paper no. CP 59-1123, presented at AIEE Fall General Meeting, Chicago, Ill., October, 1959.
[10] R. J. Nelson, "Simplest normal truth functions," *J. Symbolic Logic*, vol. 20, pp. 105–108; June, 1955.
[11] R. J. Nelson, "Weak simplest normal truth functions," *J. Symbolic Logic*, vol. 20, pp. 232–234; September, 1955.
[12] S. R. Petrick, "On the Minimization of Boolean Functions," presented at Symp. on Switching Algebra, ICIP, Paris, France; June, 1959.
[13] W. V. Quine, "The problem of simplifying truth functions," *Amer. Math. Monthly*, vol. 59, pp. 521–531; October, 1952.
[14] W. V. Quine, "A way to simplify truth functions," *Amer. Math. Monthly*, vol. 62, pp. 627–631; November, 1955.
[15] W. V. Quine, "On cores and prime implicants of truth functions," *Amer. Math. Monthly*, vol. 66, pp. 755–760; November, 1959.

General

[16] B. Dunham and R. Fridshal, "The problem of simplifying logical expressions," *J. Symbolic Logic*, vol. 24, pp. 17–19; March, 1959.

A Precision Amplitude-Distribution Amplifier*

W. F. CALDWELL†, G. A. KORN†, V. R. LATORRE†, and G. R. PETERSON†, member, ire

Summary—**A new electronic slicer circuit produces output pulses whenever a random input voltage $x(t)$ is between two slicing levels $X-\Delta x/2$ and $X+\Delta x/2$. The slicer pulses gate a counter to produce a direct digital readout count equal to the estimated first-order probability density of the input signal. The system was designed for random process studies with conventional electronic analog computers and has compatible accuracy.**

Introduction

A *slicer circuit* produces a distinct change in output voltage whenever the value of a time-variable input voltage $x=x(t)$ is between two *slicing levels*, $X-\Delta x/2$ and $X+\Delta x/2$; otherwise, the slicer output is at some quiescent level. Thus, as $x(t)$ varies, the slicer produces output pulses which may be clipped and averaged or integrated to produce an output voltage proportional to the time τ, which $x(t)$ spends between slicing levels during a fixed time interval of length T. If $x(t)$ is generated by a random process, then the fractional time constitutes an estimate of the *first-order probability density*. For greater accuracy, τ/T the pulses may gate a digital counter during a preset time interval to produce a count equal to the desired fractional time, τ/T.

The present paper describes the design and operation of a precision slicer system permitting such direct digital readout of fractional-time estimates of the probability density $p(x)$. The system was specifically designed for random-process studies with conventional real-time electronic analog computers. The use of diode switches placed in the feedback loops of standard chopper-stabilized operational amplifiers permits slicing accuracies substantially better than those previously attained [1], [2]. The first-order probability densities of random-phase sinusoids and triangular-wave voltages was estimated with an accuracy better than 1.5 per cent for an integration time of 25 periods; this accuracy is substantially that of the signal generators used. Extensions of the technique described permit estimation of second-order probability densities; and first- and second-order cumulative distribution functions could be estimated with ease.

System Design

In the block diagram of Fig. 1, output pulses from the *slicer* are inverted and applied to the *gate-pulse generator*, a bistable multivibrator. The latter produces fast-rising gate pulses in time with the slicer pulses whenever a precisely timed positive output voltage from the *timer* permits the multivibrator to operate. The gate pulses operate the *gate* so that pulses from the 1-kc *clock oscillator* are counted whenever the slicer input signal $x(t)$ is between the slicing levels. The resulting count is proportional to τ and can be made equal to the decimal fraction by adjustment of the sampling time T. Figs. 2 to 6 illustrate the operation of each block in the system.

* Received by the PGEC, August 24, 1959; revised manuscript received, February 2, 1960.
† Dept. of Elec. Engrg., University of Arizona, Tucson, Ariz.

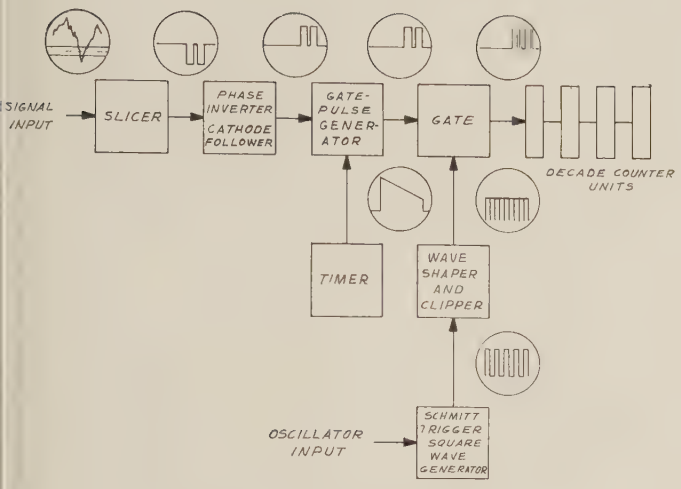


Fig. 1—System block diagram.

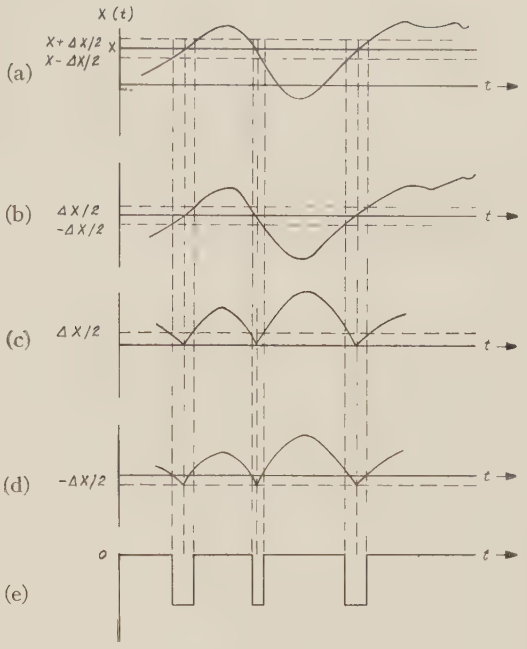


Fig. 2—Slicer operation.

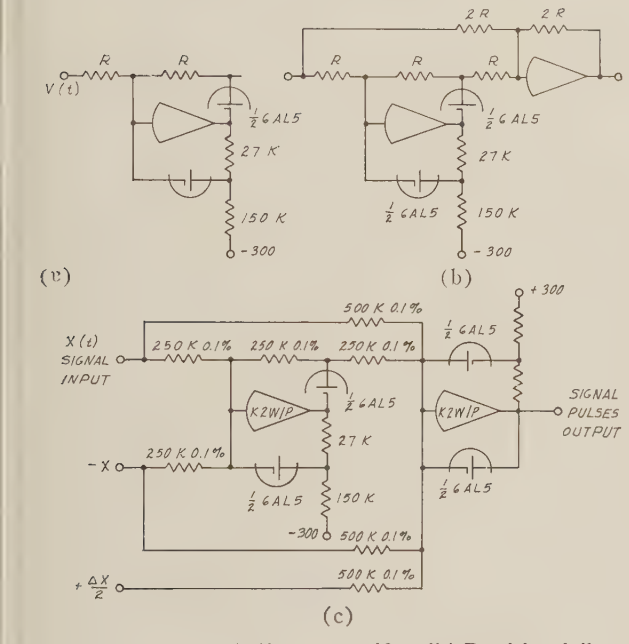


Fig. 3—(a) Precision half-wave rectifier. (b) Precision full-wave rectifier. (c) Complete slicer circuit.

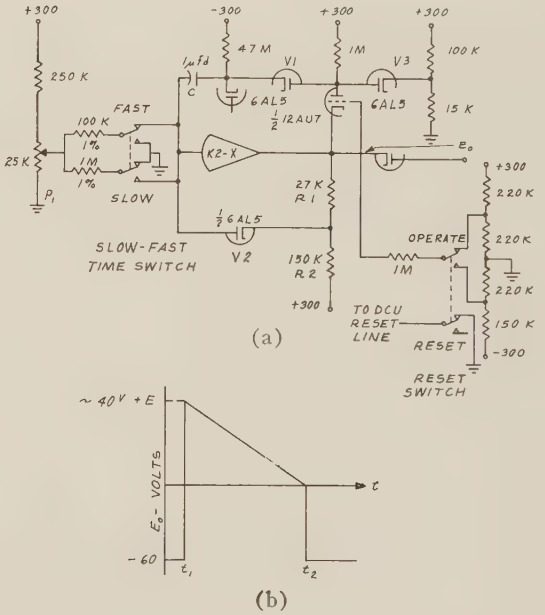


Fig. 4—(a) Precision timer schematic. (b) Output waveform.

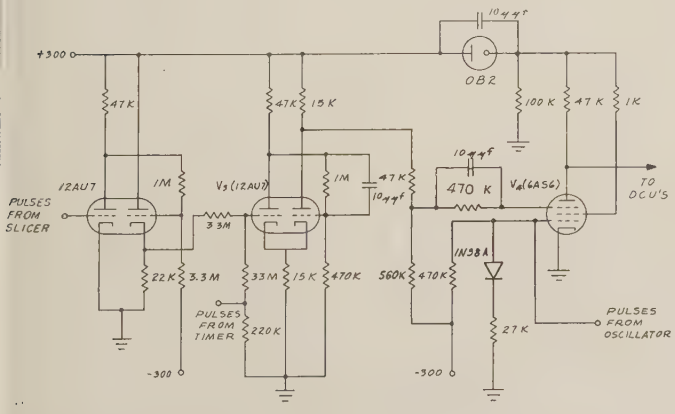


Fig. 5—Phase inverter and gate circuit.

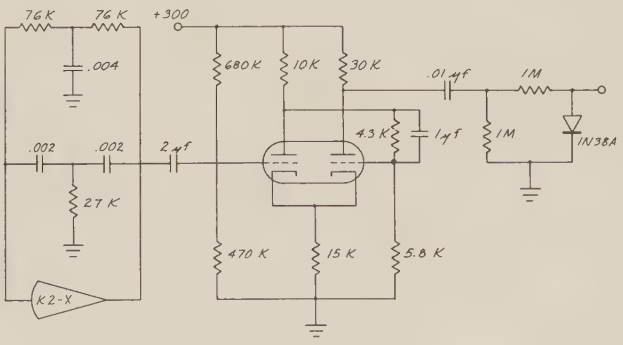


Fig. 6—Clock oscillator.

A New Slicer Circuit

The amplitude-distribution analyzer produces slicer pulses for signal-amplitude levels between $X-\Delta x/2$ and $X+\Delta x/2$ by a combination of the following operations, which are illustrated in Fig. 2:

1) The input voltage $x(t)$, shown in Fig. 2(a), is reduced by the desired amplitude level X to produce the voltage $x(t)-X$, shown in Fig. 2(b).
2) A precision full-wave rectifier rectifies $x(t)-X$ to produce the waveform of Fig. 2(c).
3) Subtraction of $\Delta x/2$ produces the waveform of Fig. 2(d), which is negative if, and only if, $X-\Delta x/2<x(t)<X+\Delta x/2$.
4) The negative portions of the last waveform activate a precision comparator amplifier, which produces the desired slicer pulses [Fig. 2(e)].

The circuits of Fig. 3 implement these operations with two operational amplifiers. The precision limiter [3], [4] of Fig. 3(a) yields linear half-wave rectification accurate to within 0.1 volt at low frequencies [5] and combines with linear circuit elements to form the precision full-wave rectifier ("absolute-value circuit") [3], [4] of Fig. 3(b).

The complete slicer circuit shown in Fig. 3(c) combines subtraction of the desired amplitude level X, precision full-wave rectification, subtraction of $\Delta x/2$, and amplification of the resulting pulses. The net input current at the summing point of the second operational amplifier has the waveform of Fig. 2(d), so that the amplifier produces output pulses in the manner of Fig. 2(e).

Timer and Resetting Circuit

The timing circuit of Fig. 4(a), developed at the University of Arizona in connection with another project [5], may be regarded as an analog-computer representation of the familiar phantastron circuit. Even without chopper stabilization, the repeatability error on consecutive runs is within 0.1 per cent, and the average repeatability error over a period of one hour is 0.5 per cent.

When the reset switch is in the RESET position, the triode is cut off, E_0 is about -60 volts, and the capacitor C charges to $+E\approx 40$ volts, as determined by the shunt limiter, V_3. At time t_1 [Fig. 4(b)], the reset switch is moved to the OPERATE position, the triode fires, and E_0 rises to $+E$ volts. With the triode conducting the circuit acts as a conventional analog integrator and E_0 decreases at a rate of a/RC volts per second, where a is the voltage at the arm of P_1, R is the input resistor, and C is the feedback capacitor [Fig. 4(a)]. When E_0 reaches 0 volts, at time t_2, diode V_1 cuts off, the gain of the system becomes essentially infinite, and E_0 moves rapidly negative until the voltage between R_1 and R_2 goes negative. At this time, diode V_2 conducts so that E_0 is limited at about -60 volts.

Thus, it is seen that the timing interval, t_2-t_1, is determined by the value of $+E$, the slope a/RC, and the value of E_0 at which V_1 cuts off. In the circuit actually used, a is adjustable by means of a multiturn potentiometer. The slow-fast-time switch yields sampling times of the order of 5 and 50 seconds.

The second contact of the reset switch resets the decimal counting units to zero.

Gate Circuits

The complete gate circuits are shown in Fig. 5. The gate-pulse-generating multivibrator (V_3) is biased so that it produces gate pulses if, and only if, both slicer pulse and timing pulse are ON. So controlled, the gate (V_4) opens when the input signal $x(t)$ is between slicing levels.

Clock Oscillator and Pulse-Shaping Circuits

A single plug-in operational amplifier with an accurately-matched double-integrator feedback network [4] becomes an accurate 1-kc oscillator [6]; no chopper stabilization is necessary, so that the oscillator is self-limiting at about ± 100 volts. The oscillator drives a bistable multivibrator (V_5), differentiating network, and clipper (1N38A) to produce fast-rising pulses for the counter (Fig. 6).

Construction

The amplitude-distribution analyzer employs standard two chopper-stabilized Philbrick K2-X/K2-P plug-in dc amplifiers and two unstabilized K2-X dc amplifiers; all these units operate with simple, regulated ± 300-volt power supplies. The decade counters are Berkeley 705AG, or similar units. The critical resistor networks in the slicer and timer circuits employ 0.1 per cent wire-wound resistors.

Operation

To use the amplitude-distribution analyzer, the desired amplitude level [$-X$ in Fig. 3(c)] and the half-slice-width $\Delta x/2$ are set on precision potentiometers connected to a suitable reference power supply. X may vary between -100 volts and $+100$ volts; the most commonly employed value of Δx is 0.5 volts for a "resolution" of 0.25 per cent of full scale. To calibrate the timer, one adjusts the sampling time T by means of potentiometer P_1 [Fig. 4(a)] so as to obtain a count of 100.0 when $x(t)=X=0$. For either operation or calibration, the reset switch is set first to RESET so as to reset timer and counters. The switch is then moved to OPERATE to initiate timer operation; the counters will read the desired estimate of the probability density in per cent.

Accuracy and Results

The voltage accuracy of the slicer and timer circuits is better than 0.1 volt at dc, and better than 0.5 volt at 100 cps, with the slicing levels set with the aid of a digital voltmeter. The timer repeatability is, as noted above, within 0.025 second or 25 counts for a time interval of 25 seconds and consecutive runs.

Rise and fall times of the slicer pulses obtained from the circuit of Fig. 3(c) are of the order of 10 μsec. Since the gate-generating multivibrator will be triggered before the slicer pulse has risen or fallen fully, the rise and fall times account for count errors of at most two counts each time the signal crosses the slicing interval. The slice width Δx must not be set to a value so low that any appreciable fraction of the slicing-level crossings take less than 20 μsec each; for longer slicer pulses, the rise and fall errors will tend to cancel. It will be desirable to scale the signal $x(t)$ so that its range lies between -100 volts and $+100$ volts.

If the above conditions are met, the analyzer will measure the desired fractional time τ/T with an accuracy better than 1 per cent; it remains to assess the value of this quantity as an estimate of probability density. Assuming that the random voltage $x(t)$ can be considered as stationary and ergodic, and that the clock-oscillator phase is statistically independent of the signal, the quality of the estimate still depends radically on the sampling time T and on the power spectrum and amplitude distribution of the signal $x(t)$; note that the quality of the estimate will, in general, vary with the level X.

If the signal $x(t)$ comprises random components of appreciable low-frequency content—at frequencies below $10/T$, for example,—then the fractional time-estimate cannot possibly give a good approximation of probability density. *In such cases, one can still use the analyzer to estimate ensemble probability densities by averaging results from repeated runs.* The range and variance of τ/T for repeated runs will, in all cases, give a measure of the quality of the estimate.

Fig. 7 shows a typical estimated probability density graph obtained with the analyzer. $x(t)$ is a 0.5-cps sinusoid. The graph shows averages of six readings with a sampling time $T=16.7$ seconds. The accuracy thus obtained was 1.5 per cent and in this case was essentially determined by the accuracy of the signal generator used.

Applications and Discussion

Applications include not only analog-computer studies of random processes, but also classroom demonstrations of the amplitude distributions of sums, products, and functions of random voltages; one may, for instance, demonstrate the central-limit theorem by analyzing the sum of a group of oscillator output voltages.

The slice width Δx can be set for any desired class interval for measurements of estimated probabilities rather than probability densities. In particular, one may estimate the cumulative distribution function

$$\Phi(X - \Delta x/2) = \text{Prob}\,[x \leq X - \Delta x/2]$$

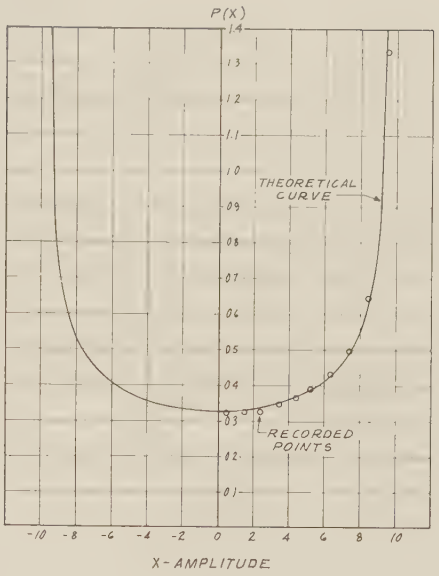


Fig. 7—Probability density of random sine wave.

of a signal whose absolute value does not exceed 100 volts; it is only necessary to set $\Delta x/2$ to a value greater than $100-X$.

Finally, addition of a second slicer and gate would permit one to estimate joint probability densities of two random processes, and, using suitable time-delay devices, the second-order probability density of a random process.

Acknowledgment

The writers wish to thank their colleague H. Koerner, who contributed many valuable suggestions. The writers are also grateful to the Electrical Engineering Department of the University of Arizona and to Dr. P. E. Russell, Department Head, for the continuing support of this project.

Bibliography

[1] D. J. Zoll, "Simple plotter analyses radar noise rapidly," *Electronics*, vol. 31, pp. 162–164; March 14, 1958.
[2] T. A. Bickart, "Amplitude slicer," *Electronics*, vol. 32, pp. 64–66; February 27, 1959.
[3] C. D. Morrill and R. V. Baum, "Diode limiters simulate mechanical phenomena," *Electronics*, vol. 25, pp. 122–126; November, 1952.
[4] G. A. Korn and T. M. Korn, "Electronic Analog Computers," McGraw-Hill Book Co., Inc., New York, N. Y. 2nd ed.,; 1956.
[5] H. Koerner and G. A. Korn, "Function generation with operational amplifiers," *Electronics*, vol. 32, pp. 66–70; November 6, 1959.
[6] G. A. Philbrick Researches, Inc., Boston, Mass., Application Sheet UPA-2.
[7] A. M. King, "An Amplitude-Distribution Analyzer," Naval Res. Lab., Washington, D. C., NRL Rept. 3890; December, 1951.
[8] E. R. Carlson, *et al.*, "Special electronic equipment for the analysis of statistical data," Proc. IRE, vol. 47, pp. 956–963; May, 1959.

A Pulse Position Modulation Analog Computer*

E. V. BOHN†, ASSOCIATE MEMBER, IRE

Summary—An important field of application for computers is in real-time systems simulation. This requires the generation of nonlinear functions, obtaining the sums and products of these functions and solving systems of nonlinear differential equations. A new type of analog computer suitable for systems simulation is described which combines the desirable features of the digital and analog computers in its mode of operation. Variables are represented by the time interval between pulses. Utilizing a few basic components, it is possible to carry out the operations of addition, subtraction, multiplication and function generation to 0.1 per cent accuracy.

Introduction

An important field of application for computers is in real-time systems simulation. This requires the generation of nonlinear functions, obtaining the sums and products of these functions and solving systems of nonlinear differential equations. A flight simulator is an example of this kind of system. A new type of analog computer suitable for systems simulation will be described. This computer has the desirable features of the digital computer in that it can handle a large number of operations with a small number of arithmetic elements and that a magnetic drum can be used for storage. It has the operational simplicity of a conventional analog computer.

The information to be handled is assigned channels on a time-sharing basis and tracks on an operation-sharing basis (see Fig. 1). This serial-parallel method of

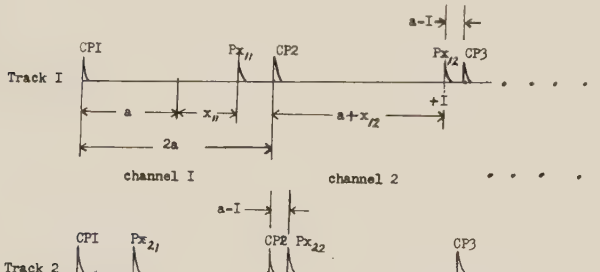


Fig. 1—Pulse position analog of computer variables.

operation enables the computer to have a relatively high speed of operation. The program for all incremental computations is carried out in one drum revolution. A variable is represented by the time interval between a fixed channel pulse CP_k and a variable pulse Px_k. This time interval is represented by $a+x$. The maximum width of a channel is then $2a$, and x may be ± 1. There is a guard band of width $a-1$ between pulses. In order to carry out arithmetic or functional operations, it is necessary to have arithmetic units which will accept these pulses in time sequence and then generate the required pulse output. This is a modulation of the pulse position by the arithmetic unit and hence the designation for this computer as a pulse position modulation analog computer. It is possible to carry out the basic arithmetic operations by a novel arrangement of precision linear sweeps.

Arithmetic Units

All arithmetic units can be built up from standard flip-flops ($FF1$, $FF2$, $\cdots$) operational amplifiers (A) and voltage comparators (C) (see Fig. 2). The basic

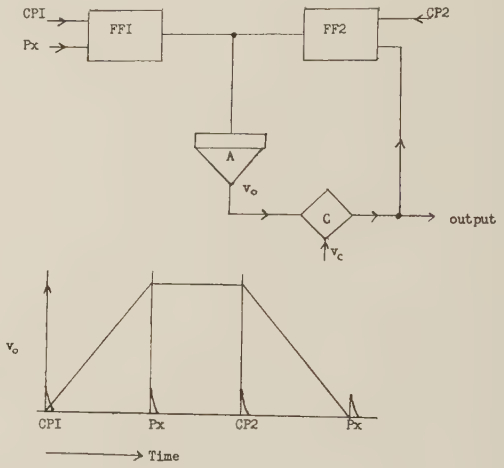


Fig. 2—Basic operation.

operation is a demodulation process which converts the pulse position to a voltage and a modulation process which converts a voltage to a pulse position. To understand this operation, consider the block diagram and the voltage-time diagram. The demodulation process is started by having channel pulse $CP1$ set $FF1$ which gates a current to the integrator A. This initiates a precision positive linear sweep which is terminated when Px resets $FF1$. The modulation process is started by having a channel pulse, for example $CP2$, set $FF2$ which initiates a downsweep. A voltage comparator C compares v_0 to a comparison voltage v_c. When $v_0 = v_c$, the comparator generates a pulse which resets $FF2$, terminating the downsweep. This comparator pulse is the output and in the case that $v_c = 0$, the output is Px.

Subtraction (Fig. 3)

This operation is based on the identity

$$2a - (a + x) = a - x.$$

* Received by the PGEC, October 28, 1959; revised manuscript received, January 18, 1960.

† Dept. of Elec. Engrg., University of British Columbia, Vancouver, B. C., Can.

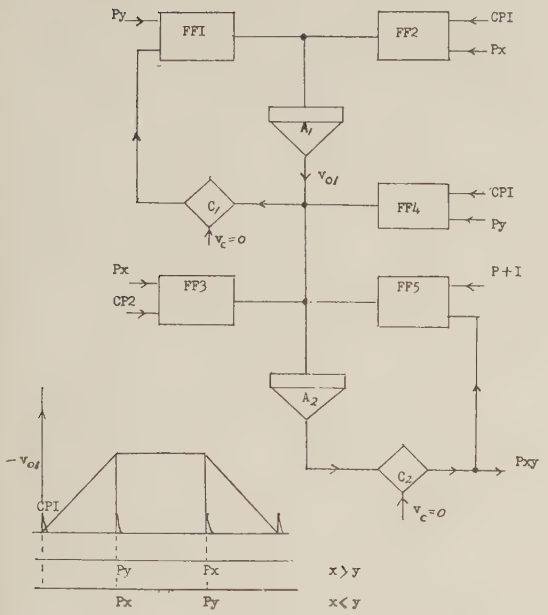


Fig. 3—Arithmetic units.

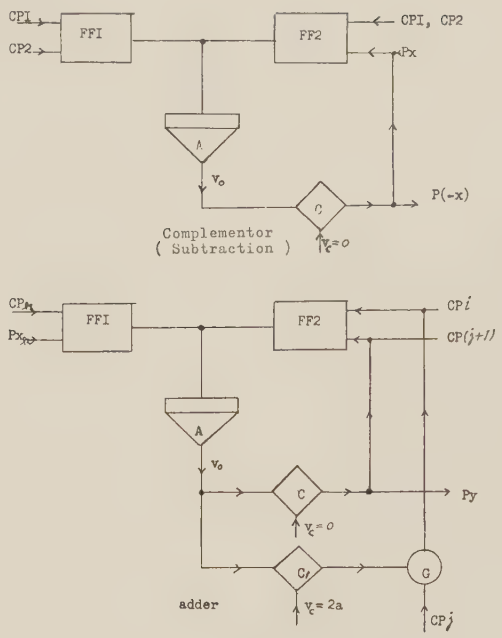


Fig. 4—Multiplier.

The $2a$ term is generated by having $CP1$ and $CP2$ set nd reset $FF1$, respectively. The $-(a+x)$ term is enerated by having $CP1$ and Px set and reset $FF2$, spectively. The result is $a-x$ as the output voltage. his is modulated by having $CP2$ initiate a downsweep hich is terminated by the comparator pulse. This ulse represents $P(-x)$.

ddition

This operation is based on the identity

$$(x_1 + a) + \cdots + (x_n + a) - a(n-1) = y + a$$

here $y = x_1 + \cdots + x_n$ is the desired sum. The ulses CP_k and x_k set and reset $FF1$, respectively. The utput represents the sum except for the $a(n-1)$ term. his is automatically subtracted by a voltage compara- or C_1 which opens the gate G whenever the output ex- eeds $2a$. This permits a channel pulse CP_j to set $FF2$. t is reset by the next channel pulse CP_{j+1}. If the number f additions is odd, this operation subtracts the required $(n-1)$. Once the sum has been obtained, a channel ulse CP_i sets $FF2$, initiating a downsweep which is erminated by the comparator pulse of C. This pulse epresents Py.

Multiplication (Fig. 4)

This operation is based on the identity

$$xy + a = (x+a)\cdot(y+a) - a(a+y) + a(a-x) - a(a-1).$$

The output of A_1 is a trapezoidal voltage v_{01} which is enerated by having $CP1$ and Px set and reset $FF2$, and aving Py set $FF1$ which is reset by the comparator C_1. his is integrated by A_2 to give the term $(x+a)\cdot(y+a)$. The next two terms are generated by having $CP1$ nd Py set and reset $FF4$, respectively, to give $-a(a+y)$ and by having Px and $CP2$ set and reset $FF3$, respectively, to give $a(a+x)$. The downsweep is initiated by having $P(+1)$ set $FF5$ which effectively gives the term $-a(a-1)$. It is reset by the comparator pulse of C_2. This pulse represents xy. The pulse $P(+1)$ is available from the computer control circuitry, which will be discussed later.

The physical condition at the input of A_1 depends on the relative value of x and y. If $y>x$, current is fed to A_1 when $CP1$ sets $FF2$ and is stopped when Px resets $FF2$. If $x>y$, the current from $FF2$ is diverted to $FF1$ when Py occurs. The output v_{01} is the same in both cases.

Function Generation[1]

Function generation is one of the most important and difficult of all the operations, since a large number of nonlinear functions usually have to be generated. Using the pulse position method, there are two practical methods of generating functions. One of these is extremely simple and its accuracy would be sufficient for many engineering problems. To understand its principle, consider Fig. 5. The functions to be generated are photographed on high-contrast 35-mm film to give transparent and opaque regions as indicated. These function frames are clamped together to form an endless band. This can be fitted to a cylinder which is rotated with the magnetic drum. A small spot of light is focused on the film and is positioned by a galvanometer to a position x. A photomultiplier tube behind the film gives an output whenever the spot strikes a transparent region. As the function frames sweep by the fixed spot, the func-

[1] F. G. Polinerou, "A new method of generating functions," IRE Trans. on Electronic Computers, vol. EC-2, pp. 29–34; September, 1954.

tions are generated in time sequence as pulse positions. The scanning rate is made sufficiently high compared to the maximum rate of change of x so that the spot remains essentially fixed during one revolution of the function cylinder. The advantage of this method over other photoelectric and cathode-ray-tube function generators is that the same equipment is used to generate a large number of functions. In addition, it is possible to use one of the function frames as a calibrating frame to correct for spot positioning errors. It is also possible to use a function frame for high-speed switching of the spot from x to another variable y so that functions of different variables can be generated with one function cylinder. Since the circuitry involved is relatively simple, it is also possible to use several function cylinders on one shaft, each with its separate circuits.

However, it is desirable to have a more flexible, precision method of function generation. A second method will now be described, utilizing a magnetic drum, which has this flexibility in modifying or changing functions rapidly. To understand this method of generating functions, consider Fig. 5, which illustrates the principles. The function to be reproduced is approximated by linear segments and parabolic arcs so that the second derivative consists of piece-wise constant values r_1, r_2, r_3, $\cdots$, with break points at x_1, y_1, x_2, y_2 $\cdots$. This gives an excellent approximation to most of the arbitrary functions occurring in physical systems, since it approximates accurately the curvature and slope of the function over selected intervals. The function is then represented by its initial values $f(0)$ and $f'(0) = df(0)/dx$ and its second derivative. To generate the function, it is only necessary to carry out a double integration. A pulse position computer is well suited for the operation. The values characterizing the function can be easily stored on a magnetic drum as pulse positions, and linear sweep circuits can perform the double integration (see Fig. 6).

The operation is started by having $CP1$ and $Pf(0)$ set and reset $FF1$, respectively. The output of A_3 represents $a+f(0)$. The reset pulse $Pf(0)$ of $FF1$ sets $FF3$ which is reset by $Pf'(0)$. The output of A_2 represents $a+f'(0)$. These are the initial values.

$FF6$ and the gates $G1$ and $G2$ control the pulses which generate d^2f/dx^2. $FF6$ is set by $CP1$, closing $G1$. It is reset by the reset pulse $Pf'(0)$ of $FF3$. This opens $G1$ and closes $G2$. The next two pulses, Pr_1 and Px_1, are routed by the sign gates $G3$ and $G4$ to set and reset $FF4$ or $FF5$. The output v_{01} represents $a+(d^2f/dx^2)$ during the interval x_1 to y_1. $FF6$ controls the gate $G5$ during this interval. The output of this gate is d^2f/dx^2. $FF6$ is then alternately set by Py_k and reset by the reset pulse Px_k of $FF4$ or $FF5$ for the remainder of the operation. $FF7$ is set by Px_1 and reset by Px. This controls the gates $G5$ and $G6$ which start and terminate the integration. A channel pulse GP_j starts the downsweep. The

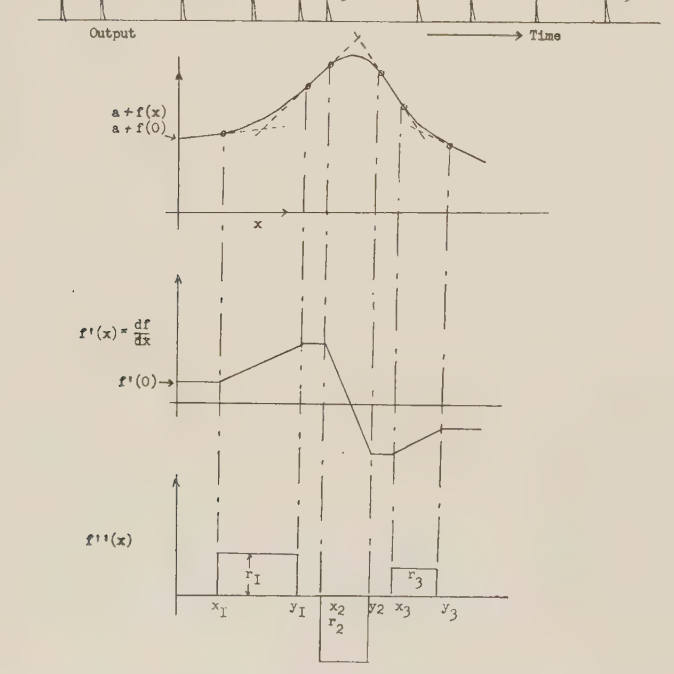


Fig. 5—Function generation.

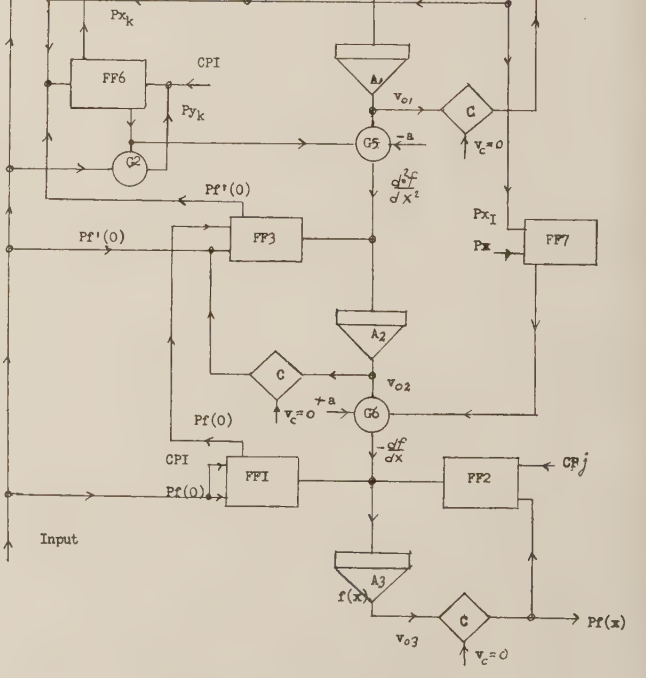


Fig. 6—Function generator.

mparator pulse represents $f(x)$. At the same time, A_1 d A_2 are cleared and the function generator is ready generate the next function.

The sign pulse is obtained by having it follow Py_k by small delay. It is detected by conventional methods d controls $G3$ and $G4$. The gates $G5$ and $G6$ are conntional current gates and are shown in Fig. 7.

The method of having all pulses follow in time seence leads to a very simple control circuit for the eration. This places a restriction on the maximum pe and curvature that can be obtained. However, the pe of functions occurring in physical systems usually eets these restrictions.

tegrator

In a pulse position computer, the integrator unit odifies the rate at which the pulse position changes. eing analog in its operation, the integrator is affected drift effects. To reduce drift to a small level, the integating action is carried out by a double-channel process. ne channel is digital in nature and is called the coarse annel. It could, for example, have 100 distinct values hich are represented by clock pulses. The integrating tion is actually carried out in an analog fine channel. he digital coarse channel simply keeps count of the mber of times the fine channel variable has reached s limits. By this method, there is no drift in the digital arse channel, and the drift of the analog fine channel comes an effect of higher order. The integrator unit ust, in order to be interconnected with other arithetic units, be built up from the same basic units.

The incremental addition in the fine scale is based on e equation (see Fig. 8):

$$\Delta y + a = (\Delta y_0 + a) + (y'\Delta t + a) - a$$

here

Δy_0 = initial value

$y'\Delta t$ = new increment (y' = derivative)

Δy = final value.

The maximum absolute value that a variable can have the fine scale is $a/2$. If the absolute value exceeds is, $a/2$ is added or subtracted from the fine scale varible and the coarse scale count decreases or increases y 1.

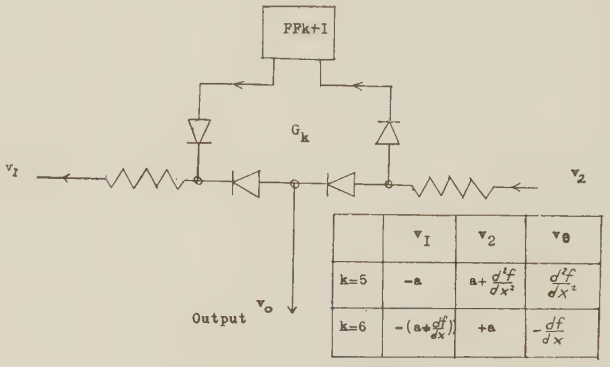
The upper and lower bounds of the fine scale are given y references voltages $v+$ and $v-$. Consider Case I. If e output voltage becomes $v+$, a gate $G1$ operates nd selects $P+$ for the downsweep. This subtracts $a/2$ om the fine scale. At the same time, 1 is added in the arse scale: Case II. The gate $G2$ selects the channel ulse. There is no change in the coarse scale. Case III: e gate $G3$ selects $P-$ which adds $a/2$ to the fine scale. t the same time, 1 is subtracted from the coarse scale.

The coarse scale operates by a pulse-selection method which eliminates drift. This is done by using quadrature clock pulses called sine clock pulses (SCP) and cosine clock pulses (CCP). Addition or subtraction is carried out by selecting a late, zero or early pulse which are separated by one unit (see Fig. 9).

Case I: Here 1 is to be added. Gate $G1$ selects a late pulse. This starts the downsweep which is terminated by the comparator. The gate $G4$ then selects the next clock pulse.

Case II: Gate $G2$ selects the zero pulse. The comparator pulse stops the sweep and opens $G4$ which selects the next clock pulse. There is no change in y_c.

Case III: Gate $G3$ selects the early pulse EP which starts the downsweep. This subtracts 1 from y_c.



	v_I	v_2	v_θ
k=5	$-a$	$a+\frac{d^2f}{dx^2}$	$\frac{d^2f}{dx^2}$
k=6	$-(a+\frac{df}{dx})$	$+a$	$-\frac{df}{dx}$

Fig. 7—Gates $G5$ and $G6$.

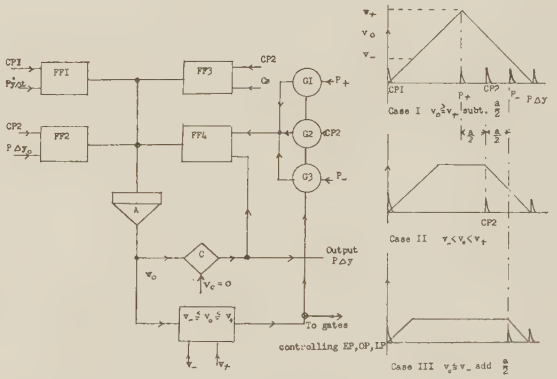


Fig. 8—Fine channel.

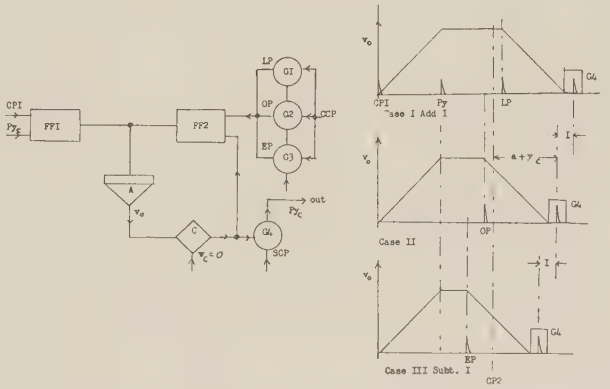


Fig. 9—Coarse channel.

Circuitry

The operational amplifier is a conventional dc type (see Fig. 10). Since the sweep is unidirectional, a transistorized automatic bias control is used. This sets the output level at about −0.2 volts, which is also the trigger level for a Schmitt-type transistor comparator. During the sweep, the change in bias is negligible. The combined use of vacuum tubes for a large range in sweep voltage with transistors for comparison of small voltages enables an accuracy of better than 0.1 per cent to be obtained for the basic sweep operation.

Computer Organization

As mentioned in the Introduction, a small magnetic drum is used to store quantities as pulse positions. A clock signal obtained from the drum is used to divide the drum circumference into 15 channels, each containing 104 clock cycles. The 104 cycles are counted down by two gas tube decade counters and four flip-flops. The two counter tubes give a quantized pulse position (1–100) which can be selected by means of two decade selector switches. This quantized pulse is used to select a phase-shifted sinusoidal signal. The zero position of this signal represents the selected pulse position.[2] The individual channels can be selected from channel gates obtained from photodiodes and a coded disk mounted on the drum (see Fig. 11).

The input-output operation is described as follows.

By means of switches, the track and channel position of the drum is selected. The two decade selector switches and the phase shifter are positioned. A manual push-button switch is used to start the pulse selection operation described and the pulse is written on the required track and channel. For readout, a null method is used to compare the pulse position in the selected track and channel with the selector pulse. The obtainable accuracy for a single input-output operation is very high, the error being less than 0.03 per cent of full scale.

The systems to be simulated can be represented by systems of differential equations of the type:

$$\frac{du_j}{dt} = \sum_{k=1}^{m} \{f_k(x)g_k(y) + h_k(z)\} + \text{external inputs}$$

$$(j = 1, \cdots, n). \quad (1)$$

Considering the time t to be a parameter, the x, y, z represent the independent variables. (In aircraft simulation, these could be the mach number, altitude and true airspeed.) The u_j represent x, y, z and their derivatives and all other dependent variables.

The system organization to simulate (1) is shown in Fig. 11. The computer program is controlled by a patchboard which connects the desired sequence of channel gates to the various arithmetic and functional units. The selected channel gate (Chg.) selects a channel pulse which sets a flip-flop. It is reset by the next variable pulse (see Fig. 12). The pulse code for the functions to be generated and the scaling coefficients and initial conditions for the integration are entered into selected tracks and channels on the drum. The function generators generate the function $f_k(x)$, $g_k(y)$ and $h_k(z)$ in a prescribed time sequence, and the multiplier generates the products $f_k(x)g_k(y)$. The adders are controlled by selected channel gates and generate the sums in (1). Whether one adder can generate more than one sum depends on the time available. In general, a number of adders are required. At a prescribed time, the adder is cleared. The sum enters a multiplier for scaling and then goes into the integrator for an incremental addition. The output of the integrators represents the new values.

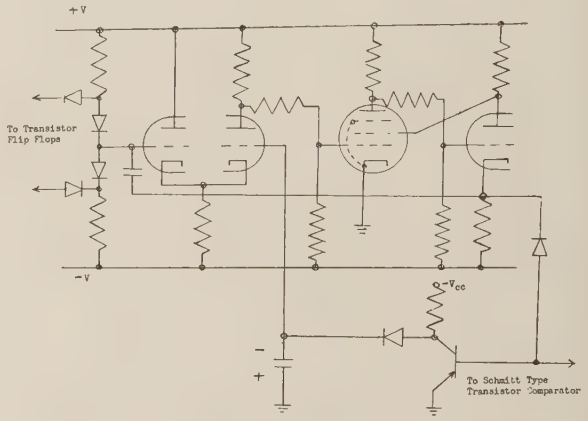


Fig. 10—Operational amplifier.

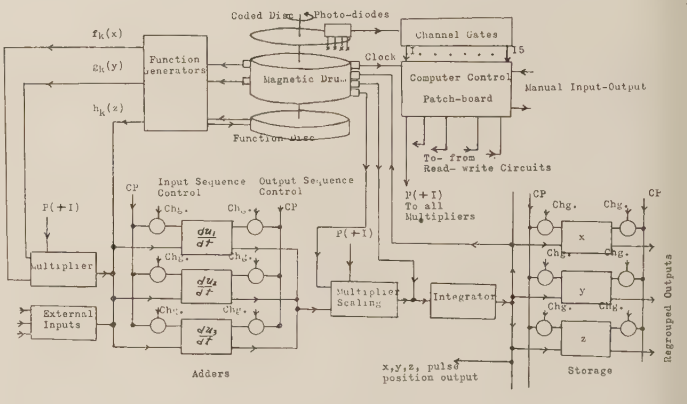


Fig. 11—Computer organization.

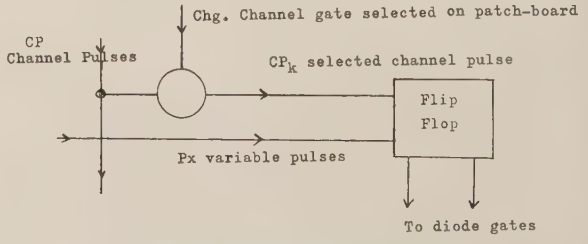


Fig. 12—Program control gates.

[2] J. Millman and H. Taub, "Pulse and Digital Circuits," McGraw-Hill Book Co., Inc., New York, N. Y., pp. 499–501; 1956.

or all variables. Since an incremental computation is arried out in each drum revolution, there must be sufcient serial-parallel operations available to complete he entire program in this time. This also means that the equence of outputs from the integrator has to be regrouped in order to operate the function generators in he proper time sequence. A simple method of regrouping is to use adders as a storage for all independent ariables and to code these into the proper time sequence by means of selected channel gates.

Conclusion

Computing elements based on these principles have been built and tested. The circuitry is simple and reliable, and accuracies in the range of 0.1 per cent to 1 per cent have been obtained with a time interval $a = 1000$ μsec. The results to date indicate that an accuracy comparable to the best obtainable in present analog computing elements can be achieved and possibly even surpassed. This requires precision calibration of sweep voltages and time constants. These features are presently being introduced. The over-all performance as a computer still remains to be tested.

It is interesting to compare a pulse position computer with other developments in the field of systems simulation. Ultra-high-speed digital[3,4] and digital differential analyzers[5] are costly and the very high computational accuracy is not always required. The method of integration used in the pulse position computer is similar to the incremental computation technique of a digital differential analyzer. Hybrid computers[6–8] are complex and represent a compromise between cost, speed and accuracy.

In comparison, it is seen that a pulse position computer represents a new analog method of achieving the compromise of the hybrid computer. Its computational technique can be readily supplemented by conventional digital or analog methods since its information is readily converted to a digital code or analog voltage.

This computer has several important advantages. It uses time intervals directly which is the only dynamic quantity suitable for high-accuracy analog computation. This makes it possible to carry out a step by step incremental computation with a very accurate analog readout. The measurement of the time varying voltage of a conventional analog computer to the same accuracy requires very elaborate circuitry. It is very simple to time share computing elements by gating pulses. Time sharing in a conventional analog computer requires the much more difficult operation of voltage gating as is the case of the computers described by Lee and Cox[7] and Herzog.[8]

Acknowledgment

The author gratefully acknowledges the generous support given to this project by the Defence Research Board of Canada under grant No. 9931–02 and to the several graduate students who worked on the development of the associated circuitry.

[3] W. Dunn, C. Eldert, and P. Levonian, "A digital computer for use in an operational flight trainer," IRE Trans. on Electronic Computers, vol. EC-4, pp. 55–63; June, 1955.

[4] A. Ashley, "A five microsecond memory for UDOFT computer," 1957 WESCON Convention Record, pt. 4, pp. 262–266.

[5] J. Mitchell and S. Ruhman, "The Trice—a high speed incremental computer," 1958 IRE National Convention Record, pt. 4, pp. 206–209.

[6] R. Leger and J. Greenstein, "Simulate digitally, or by combining analog and digital computing facilities," *Control Engrg.*, vol. 3, pp. 145–153; September, 1956.

[7] R. Lee and F. Cox, "A high speed analog digital computer for simulation," IRE Trans. on Electronic Computers, vol. EC-8, pp. 186–196; June, 1959.

[8] A. Herzog, "Pulsed analog computer for simulation of aircraft," Proc. IRE, vol. 47, pp. 847–851; May, 1959.

Correction

George W. Reitwiesner, author of "The Determination of Carry Propagation Length for Binary Addition," which appeared on pages 35–38 of the March, 1960, issue of these Transactions, has called the following to the attention of the Editor.

On page 38, in Table I, two entries are in error, as follows:

1) For $m=4$, $j=3$, $k=3$, insert entry $\overline{Q}_{4,3,3}=2$.
2) For $m=4$, $j=4$, $k=3$, the entry 1 should be deleted.

On page 128, in the biography, the author was employed at the Ballistic Research Laboratory from 1947 to 1959, not the year 1949.

Correspondence

A Note on Magnetic Shift Registers*

A good deal of research has been carried out at Stanford Research Institute on a type of multi-aperture-device shift register employing absorption of flux in coupling-loop resistance during part of the transfer cycle. A recent article[1] by Gianola includes some results of independent research carried out at Bell Laboratories on the same technique. This note is for the purpose of pointing out an apparent error in Gianola's article, for showing that his chosen turns ratio is optimum in a sense, and for discussing drive tolerances briefly. Papers planned for presentation (by myself and David Nitzan) at the 1960 Non-Linear Magnetics and Magnetic Amplifiers Conference will provide more complete coverage of the work at Stanford Research Institute in this area.

On page 330 of Gianola's article, in connection with an earlier type of register,[2-4] it is stated that

$$\dot{\phi}_2 \rightarrow n\dot{\phi}_1 \quad \text{and} \quad i \rightarrow \frac{2nI + I_0}{1 + n^2} \tag{1}$$

as $R \rightarrow 0$, where R = coupling-loop resistance; $\dot{\phi}_1$ and $\dot{\phi}_2$ are the rates of flux change in cores 1 and 2 (transmitter and receiver), respectively; i = induced loop current; I = drive current; I_0 = main-aperture threshold current; and n:1 is the turns ratio. Based on the first part of (1), the excess field applied to core 2, which is proportional to $(i - I_0)/L$, should in the limit be n times the net excess field acting on core 1, which is proportional to

$$\frac{2I - ni - (l/L)I_0}{l}$$

where l and L are path lengths around the minor and main apertures, respectively. Hence,

$$\frac{i - I_0}{L} \rightarrow n\,\frac{2I - ni - (l/L)I_0}{l}. \tag{2}$$

Solution of (2) for i yields

$$i \rightarrow \frac{2nIL - (n-1)I_0 l}{l + n^2 L}. \tag{3}$$

Eq. (3) is suggested as a replacement for the second part of (1), which appears to be in error. The next equation of Gianola, with an apparent typographical error corrected, is

$$\frac{1}{\tau} = \frac{4\pi}{s_w L}(i - I_0) \tag{4}$$

(in electromagnetic units), where τ is the transfer time (the main limitation on speed) and s_w the switching coefficient. Elimination of i from (3) and (4), followed by replacement of I with its maximum value $I_0 = H_0 L/4\pi$, results in

$$\tau \rightarrow \frac{s_w}{H_0}\left(\frac{l + n^2 L}{2nL - nl - n^2 L}\right) \tag{5}$$

as $R \rightarrow 0$. If we now let $n \rightarrow 1$ and l/L go to zero, then

$$\tau \rightarrow \frac{s_w}{H_0} \tag{6}$$

in the limit, whereas the corresponding equation of Gianola yields

$$\tau \rightarrow 2\,\frac{s_w}{H_0} \tag{7}$$

in the limit as $n \rightarrow 1$. However, even (6) does not represent the best that can be done. If the drive winding is also made to link leg 1 of the receiver (core 2), the latter will be biased to threshold by the drive current $I = I_0$. Then (2) through (6) may be replaced in order, by

$$n\,\frac{2I - ni - (l/L)I_0}{l} = \frac{i}{L}, \tag{8}$$

$$i = \frac{2nIL - nlI_0}{l + n^2 L}, \tag{9}$$

$$\frac{1}{\tau} = \frac{4\pi}{s_w L}\,i, \tag{10}$$

$$\tau = \frac{s_w}{H_0}\left(\frac{l + n^2 L}{2nL - nl}\right), \tag{11}$$

$$\tau \rightarrow \frac{1}{2}\,\frac{s_w}{H_0}. \tag{12}$$

For the type of register described by Gianola, (6) on his page 329 indicates that

$$\tau_p \rightarrow \frac{s_w}{H_0}\,\frac{n^2}{n-1} \tag{13}$$

as $l/L \rightarrow 0$, where τ_p is the priming time (the main limitation on speed in this case). It is stated by Gianola that n should be kept as small as is compatible with gain requirements. Actually, the minimum τ_p is obtained with $n = 2$ (as will presently be shown), in which case

$$\tau_p \rightarrow 4\,\frac{s_w}{H_0}. \tag{14}$$

However, as Gianola indicates on page 313, an additional winding (E_2) can increase the speed up to a factor of two. Hence

$$\tau_p \rightarrow 2\,\frac{s_w}{H_0} \tag{15}$$

represents the actual lower limit on τ_p. Comparison of (15) with (12) shows that this system has one-fourth the basic speed potentiality of the first one discussed here. However, the greater drive current tolerances obtained in exchange are very significant.

The lower limit on τ_p given by (13) may be written in the form

$$\tau_p(n) = K\,\frac{n}{n-1} \tag{16}$$

where K is a constant. Differentiation of τ_p with respect to n shows that $\tau_p(n)$ is minimum when $n = 2$. Hence, there is more reason for choosing this value than just convenience, the reason given on page 308 of Gianola. Furthermore, note that

$$\tau_p\left(\frac{n}{n-1}\right) = K\,\frac{n^2/(n-1)^2}{\dfrac{n}{n-1} - 1} = K\,\frac{n^2}{n-1}.$$

That is,

$$\tau_p\left(\frac{n}{n-1}\right) = \tau_p(n). \tag{17}$$

In other words, the same lower limit on τ_p will be obtained with a turns ratio of 3:1 as with one of 3:2, the same with 4:1 as with one of 4:3, etc., and all of these lower limits will be greater than the one resulting from

$$n = \frac{n}{n-1} = 2.$$

The upper limit on R and the total drive power required will be greater, however, for turns ratio n:1 than for ratio n:$n-1$, assuming $n > 2$.

On page 311 of Gianola, it is implied by the statements relative to margins on drive current A that there is no upper limit on A. Such may appear to be the case since current A can only push a transmitting core containing a *zero* further into saturation. However, purely elastic flux changes coupled into the transfer loop can cause loop current to exceed the threshold of the receiver and start buildup of flux from a *zero* level to a *one* level, if the amplitude of A becomes sufficiently large, given a fixed rise time. To put it another way, for large values of peak current A, there tends to be an upper limit on dI/dt on the pulse front, beyond which *zero* buildup will occur.

A final comment, in support of Gianola's indications of the practicality of the scheme, seems in order. Obviously, required drive-current levels will depend upon coupling-loop resistance values, and hence the question of uniform drive-current requirements of a batch of production registers arises. The figures given below were supplied by AMP, Inc., upon my request for an indication of the degree of uniformity presently being obtained. For a production run of 250 ten-bit registers, driven by round-topped transfer pulses of 1.5 ampere peak amplitude, the minimum allowable priming current was found to have an average value of 71 ma with ± 9 per cent variation over the whole batch. The maximum allowable priming current had an average value of 254 ma with less than ± 6 per cent variation for 248 out of the 250 registers.

D. R. Bennion
Stanford Res. Inst.
Menlo Park, Calif.

* Received by the PGEC, April 26, 1960. The Stanford Res. Inst. research referred to in this note is being sponsored by AMP, Inc., of Harrisburg, Pa.

[1] U. F. Gianola, "Integrated magnetic circuits for synchronous sequential logic machines," *Bell Sys. Tech. J.*, vol. 39, pp. 295–332; March, 1960.

[2] H. D. Crane, "A high speed logic system using magnetic elements and connecting wire only," *Proc. Special Technical Conf., Nonlinear Magnetics and Magnetic Amplifiers*, Los Angeles, Calif., pp. 465–482; August, 1958.

[3] N. S. Prywes, "Diodeless magnetic shift registers utilizing transfluxors," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 316–324; December, 1958.

[4] D. R. Bennion and H. D. Crane, "Design and analysis of MAD transfer circuitry," *Proc. WJCC*, pp. 21–36; March, 1959.

omments on a Character ecognition Method of Bledsoe d Browning*

A method of Bledsoe and Browning[1] for aracter recognition was tested at the Bell lephone Laboratories, using the General-d Scanner[2] as an input transducer and the M 704 to simulate the recognition logic. e recognition of both hand block printing d machine printing was studied.

The hand-printed data consisted of 50 habets of 36 characters, each alphabet nted by a different person. The printing s somewhat constrained by requiring ese people to print on one-quarter-inch ad-ruled paper and asking them to print atly and at a size approximating the ruled xes on the paper. Thus, a total of 1800 niconstrained alpha-numeric characters re studied.

The source of the machine printing was IBM 407 line printer. Recognition of samples of each of the ten machine-nted numerals was attempted.

Recognition for this method was tried randomly chosen ordered exclusive pairs ing a 12×12 scanning matrix and five mples to construct the "memory matrix." ntering was done in the same manner as ggested by Bledsoe and Browning; that is, e input pattern was shifted to the upper t-hand corner of the matrix.

It was felt that any results obtained with eir method would be meaningless unless ey were compared with the results of an-her recognition method operating on the me data.

The recognition method to which the sults of Bledsoe and Browning's method re compared involves the comparison of unknown input pattern (the same 12×12 trix is used in both methods) to a set of erage characters. The average characters e described by a set of 12×12 matrices ne for each character) in which each ele-ent represents the probability of occurrence a mark in that element for the character ich it represents. For example, there is e average character for A, one for B, and on. The comparison of the unknown in-t pattern to the average characters is per-rmed by computing a set of cross-correla-on values, and the maximum value is osen as a criterion for identification. The put pattern is shifted in two dimensions th respect to the average characters to ac-unt for off-center characters.

In both methods, letters were compared ly to letters and numbers to numbers. ne results are shown in Table I. The re-lts obtained by Bledsoe and Browning are ken from their paper, and the results using eir method are for samples of each charac-r which were not used to make up the list. ne results for the correlation method are sed on all 50 samples. In the case of hand inting, the samples recognized were the same samples which were used to construct the average characters; in the case of machine printing, recognition was tried for a different set of 50 samples of each numeral from that used to construct the probability matrices.

TABLE I

PER CENT RECOGNITION FOR VARIOUS TRIALS

	Results Obtained by Bledsoe and Browning	Duplication of Method of Bledsoe and Browning	Results of Correlation Method
Hand printing	78.4	19.6	77.2
Machine printing	100.0	86.7	99.6

It is evident that this study does not verify the results of Bledsoe and Browning. For the case of hand printing, the differences may be ascribed to the fact that their input data were generated by only one person rather than by a representative population.

W. H. HIGHLEYMAN
L. A. KAMENTSKY
Bell Telephone Labs.
Murray Hill, N. J.

* Received by the PGEC, April 11, 1960.
[1] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *Proc. EJCC*, pp. 5–232; December, 1959.
[2] W. H. Highleyman and L. A. Kamentsky, "A neralized scanner for pattern- and character-ognition studies," *Proc. WJCC*, pp. 291–294; arch, 1959.

Improved Arrangement of a Decimal Multiplier*

Dr. Phister described a decimal multiplier[1] in his book, "Logical Design of Digital Computers." The product of two decimal digits is formed by adding, in a three-input adder, the proper combination of the multiplicand multiples (Y), ($-Y$), and ($4Y$). The ($-Y$) multiple is formed by the 10's complement of the multiplicand. A table is presented on page 307 of the book mentioned above which lists a combination for each of the nine possible multiples to be added. There are several other variations to the table which Dr. Phister does not list. With a rearrangement of the $2Y$ and $3Y$ adder input combinations, a reduction of better than one-third in the number of diodes necessary to build the adder inputs can be easily accomplished in a binary coded decimal machine.

The rearranged table is as follows.

Desired Product	The Adder Inputs		
	X_1	X_2	X_3
Y	0	Y	0
$2Y$	0	Y	Y
$3Y$	0	$-Y$	$4Y$
$4Y$	$4Y$	0	0
$5Y$	$4Y$	Y	0
$6Y$	$4Y$	Y	Y
$7Y$	$4Y$	$-Y$	$4Y$
$8Y$	$4Y$	0	$4Y$
$9Y$	$4Y$	Y	$4Y$

To realize the reduction in number of diodes necessary to build this multiplier, let us look at a simple example. Two 8-4-2-1 decimal coded digits are to be multiplied by the above table. The multiplicand is contained in a register and the multiplier in a register, $R_1R_2R_3R_4$. Multiplication shall be done in one bit time, hence in a parallel-by-digit mode. Whether operating in parallel or serially, the diode reduction is the same. This is true because the multiple combinations, which are a function of the multiplier digits, must be determined before the multiplication of the multiplicand commences. The new adder input equations are

$$X_1 = Z_4R_4 + Z_4R_3 \text{ (6 diodes)},$$

$$X_2 = Z_1R_1\bar{R}_2 + Z_1\bar{R}_1R_2 + Z_{-1}R_1R_2 \text{ (12 diodes)},$$

$$X_3 = Z_4R_4 + Z_1\bar{R}_1R_2 + Z_4R_1R_2 \text{ (11 diodes)}.$$

In these equations, Z_4 means the output of the $4Y$ multiple generator, Z_1 means the multiplicand (Y) delayed in a serial machine, and Z_{-1} is the ($-Y$) multiple or the complemented multiplicand.

The number of diodes required to make up these adder input gates can be compared with the number required to make up the equations on page 311 of Phister's book. The author calculates that 44 diodes are necessary to construct the adder inputs of Dr. Phister's multiple combination scheme. Using the same gate construction rules, he calculates that only 29 diodes are necessary to construct the adder inputs of the multiple combination scheme described in this letter.

JOHN E. CROY
ElectroData Div.
Burroughs Corp.
Pasadena, Calif.

* Received by the PGEC, April 11, 1960.
[1] M. Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y., pp. 305–311; 1958.

Computation of an Expression for the Frequency Spectrum of a Two-Sided Signal*

For the purpose of this discussion we will define a two-sided signal, $s(t)$, to be one which is not identically zero and which is of class L_1 everywhere within the open interval $(-\infty, +\infty)$. The frequency spectrum, $S(\omega)$, of this signal is conventionally determined by taking the Fourier transform of $s(t)$ as:

$$S(\omega) = \int_{-\infty}^{+\infty} s(t)\epsilon^{-j\omega t}dt. \qquad (1)$$

If $S(\omega)$ is determined by machine, then it is calculated for a large number of values of ω to yield finally a tabular form for $S(\omega)$. It is to be noted further that a machine calculation can, in general, carry out an integration in the variable t only over a finite interval, say $(-T, +T)$. Thus, the machine would,

* Received by the PGEC, December 2, 1959; revised manuscript received, February 1, 1960.

in actuality, be determining only the tabular values of a $\tilde{\tilde{S}}(\omega)$, where

$$\tilde{\tilde{S}}(\omega) = \int_{-T}^{+T} s(t)\epsilon^{-j\omega t}dt. \tag{2}$$

It is, of course, assumed that a judicious choice of T will result in

$$E - \tilde{E} < \epsilon \tag{3}$$

where E and $\tilde{E}$ are the values of the energies in the signals as given by

$$E = \frac{1}{2\pi}\int_{-\infty}^{+\infty} |S(\omega)|^2 d\omega = \int_{-\infty}^{+\infty} |s(t)|^2 dt \tag{4}$$

$$\tilde{E} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} |\tilde{\tilde{S}}(\omega)|^2 d\omega = \int_{-T}^{+T} |s(t)|^2 dt. \tag{5}$$

The value of ϵ is to be arbitrarily small and specifiable. Defined in this way, $\tilde{\tilde{S}}(\omega)$ can be used to approximate $S(\omega)$ in the mean. It will be tacitly assumed in the following discussion that for the $s(t)$ of interest, a T can be found which satisfies condition (3). The point to be made is that all of these calculations will, at best, lead only to a tabular form for $\tilde{\tilde{S}}(\omega)$ and not to an explicit functional form for $\tilde{\tilde{S}}(\omega)$. This can be remedied.

The set of Hermite functions $\{h_n(t)\}$[1] forms a complete set and can, therefore, be used to give a series expansion for $s(t)$, as

$$s(t) = \sum_{n=0}^{\infty} a_n h_n(t) \tag{6}$$

where

$$a_n = \frac{1}{2^n n!\sqrt{\pi}}\int_{-\infty}^{+\infty} s(t)h_n(t)dt. \tag{7}$$

Now if the series (6) can be integrated term by term, then

$$S(\omega) = \sum_{n=0}^{\infty} a_n H_n(\omega) \tag{8}$$

where $H_n(\omega)$ is the Fourier transform of $h_n(t)$. The simplicity in this representation for $S(\omega)$ comes about because of the following relationship:[2]

$$H_n(\omega) = \sqrt{2\pi}(-j)^n h_n(\omega). \tag{9}$$

Thus,

$$S(\omega) = \sqrt{2\pi}\sum_{n=0}^{\infty} (-j)^n a_n h_n(\omega). \tag{10}$$

For the purposes of a machine calculation, the integration range for t in (7) would have to be restricted to a finite interval, say $(-T, +T)$. This would yield

$$\tilde{a}_n = \frac{1}{2^n n!\sqrt{\pi}}\int_{-T}^{+T} s(t)h_n(t)dt \tag{11}$$

[1] The $h_n(t)$ can be expressed as:

$$h_n(t) = \epsilon^{-t^2/2}\left[(-1)^n\epsilon^{+t^2}\frac{d^n}{dt^n}\epsilon^{-t^2}\right]; n = 0, 1, \cdots$$

[2] M. I. Aissen, "Hermite Polynomials and Functions," Rad. Lab., The Johns Hopkins University-Baltimore, Md., Rept. No. RL/58/IMA-6; 1958.

and

$$\tilde{\tilde{s}}(t) = \sum_{n=0}^{\infty} \tilde{a}_n h_n(t)$$
$$= s(t) \quad -T \le t \le +T$$
$$= 0 \quad t < -T, t > +T. \tag{12}$$

Corresponding to $\tilde{\tilde{s}}(t)$ there is an $\tilde{\tilde{S}}(\omega)$ given in series form by:

$$\tilde{\tilde{S}}(\omega) = \sqrt{2\pi}\sum_{n=0}^{\infty} (-j)^n \tilde{a}_n h_n(\omega). \tag{13}$$

$\tilde{\tilde{S}}(\omega)$ gives an approximation to $S(\omega)$ in the sense described by (3). The energies, E and $\tilde{E}$, can now be written in series form as

$$E = \sum_{n=0}^{\infty} [2^n n!\sqrt{\pi}]\,|a_n|^2 \tag{14}$$

$$\tilde{E} = \sum_{n=0}^{\infty} [2^n n!\sqrt{\pi}]\,|\tilde{a}_n|^2. \tag{15}$$

The actual machine approximation to $S(\omega)$ will not be as good as indicated by (3), since, in addition to taking only a finite integration interval, it will be possible to calculate only a finite number of the $\tilde{a}_n$, say the first N of them. Thus, there will be a $\tilde{\tilde{\tilde{S}}}(\omega)$ defined as

$$\tilde{\tilde{\tilde{S}}}(\omega) = \sqrt{2\pi}\sum_{n=0}^{N} (-j)^n \tilde{a}_n h_n h_n(\omega). \tag{16}$$

For N sufficiently large,[3]

$$\tilde{\tilde{E}} - \tilde{E} < \epsilon \tag{17}$$

where

$$\tilde{\tilde{E}} = \sum_{n=0}^{N} [2^n n!\sqrt{\pi}]\,|\tilde{a}_n|^2. \tag{18}$$

This then, insures that

$$\tilde{\tilde{E}} - \tilde{\tilde{E}} < 2\epsilon. \tag{19}$$

The result of all of this is that by choosing the Hermite functions to represent the two-sided signal $s(t)$, a simple, *functional* approximation in the mean to the frequency spectrum $S(\omega)$ can be obtained by machine calculation.

$$S(\omega) \approx \sqrt{2\pi}\sum_{n=0}^{N} (-j)^n \tilde{a}_n h_n(\omega) \tag{20}$$

$$\tilde{a}_n = \frac{1}{2^n n!\sqrt{\pi}}\int_{-T}^{+T} s(t)h_n(t)dt. \tag{21}$$

It is further conceivable that there will be many functions, $s(t)$, for which far fewer $\tilde{a}_n$ would have to be determined than points in a point-by-point tabulation of $S(\omega)$ using the Fourier transform; this, of course, results in a saving of expensive computing time.

THEODORE A. BICKART
The Johns Hopkins University
Baltimore, Md.

[3] The value of N can be specified by utilizing the mean-square-error criterion and the orthogonality property of the Hermite functions as:

$$\int_{-T}^{+T} |s(t)|^2 dt - \sum_{n=0}^{N} [2^n n!\sqrt{\pi}]\,|\tilde{a}_n|^2 < \epsilon.$$

Self-Complementary Symmetry Types of Boolean Functions*

This note is concerned with those Boolean functions (switching functions) that are of the same symmetry type as their binary complements. Such symmetry types are called *self-complementary*.

The Pólya-Slepian enumeration formula for symmetry types is modified to permit one to count the number of such self-complementary symmetry types for an arbitrary number of variables, n. Using the modified formula, this number has been computed for $n=1$, 2, 3, 4, and 5. For $n=1$, 2, and 3 all *neutral* symmetry types (those with 2^{n-1} "true" entries in the truth table) are self-complementary. When n is greater than three, this relation breaks down and the self-complementary types become increasingly rare.

A network interpretation is given of Slepian's classification of variable transformations into *e-cycles* and *o-cycles* which sheds light on the problem of classifying Boolean functions and relates this problem to sequential network theory.

Boolean functions, $f(x_1 \cdots x_n)$ and $g(x_1 \cdots x_n)$, are said to belong to the same symmetry type if there exists some variable transformation (permutation and/or complementation of some or all of the variables $x_1, \cdots, x_n$) which changes f into g. Tables[1,2] of symmetry types exist through $n=4$ and the numbers of distinct symmetry types have been calculated[3] up to $n=6$.

Suppose that $f=f(x_1 \cdots x_n)$ is a Boolean function of n variables and that $\bar{f}=\bar{f}(x_1 \cdots x_n)$ is its binary complement. In order for f and f to belong to the same symmetry type, it is clearly necessary that f (and, hence, also $\bar{f}$) have value *1* for 2^{n-1} of the 2^n possible input combinations, and have value *0* for the other 2^{n-1} combinations. Such functions (having equal occurrences of value *1* and value *0*) have been called *neutral*[4] and they have important properties in relation to the study of sequential networks.[4]

Consideration of the possible symmetry types of neutral functions readily shows that for $n=1$, 2, and 3, all neutral functions belong to the same symmetry type as their respective binary complements, and, hence, are self-complementary. The familiar parity functions are all of this type. Other examples of self-complementary symmetry types are shown in Fig. 1. It should not be thought, however, that neutral symmetry types are invariably self-complementary. Of the 72 neutral symmetry types in four variables, 42 turn out to be self-complementary while the remaining 32 occur in complementary pairs (see Higgonet and Grea[2]). Example

* Received by the PGEC, February 23, 1960; revised manuscript received, May 9, 1960.

[1] "Synthesis of Electronic Computing and Control Circuits," The Staff of the Computation Laboratory, Harvard University, Harvard University Press, Cambridge, Mass.; 1951.

[2] R. A. Higgonet and R. A. Grea, "Logical Design of Electrical Circuits," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958. See "Table of Four-Relay Circuits," by E. F. Moore, pp. 195–216.

[3] D. Slepian, "On the number of symmetry types of Boolean functions of n variables," *Can. J. Math.*, vol. 5, no. 2, pp. 185–193; 1953. Also Bell Telephone System Monograph 2154.

[4] W. H. Kautz, "State-logic relations in autonomous sequential networks," *Proc. EJCC*, Boston, Mass.; December, 1958.

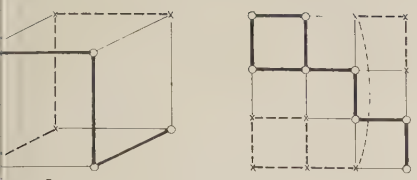
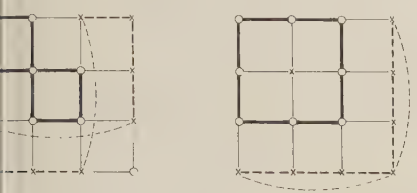


Fig. 1—Examples of self-complementary neutral symmetry types.



2—Karnaugh diagrams of some non-self-complementary neutral symmetry types; $n=4$.

shown in Fig. 2 by means of Karnaugh grams.[5]

The question thus arises as to what hapns for $n>4$. How does the class of neutral ctions divide into self-complementary C) and non-self-complementary (NSC) es? Do both SC and NSC types exist all values of $n\geq 4$? These questions have en partially answered by an extension of pian's methods.[3]

The basic tool in this investigation is the mula (Pólya,[6] Slepian[3])

$$N_n = \frac{1}{n!2^n}\sum_C n_C 2^{K(C)} \qquad (1)$$

the number of distinct symmetry types Boolean functions in n (or fewer) varies. Here the summation is extended over classes C of equivalent operations (variable transformations), with n_C the number operations in class C while $K(C)$ is the mber of cycles into which the vertices of e n-dimensional hypercube are permuted an operation of class C. Thus, $2^{K(C)}$ es the number of Boolean functions inriant under an operation of class C. For more detailed explanation of these conpts, the reader should refer to Slepian's per.[3]

As pointed out by Golomb,[7] (1) is an stance of a very general expression for the mber of transitivity sets into which a ace S is decomposed by the operations of finite transformation group G acting on e points of this space. Two points, s and s', e in the same transitivity set if, and only some operation t of G transforms s into s', , if $t(s)=s'$. Let $I(t)$ be the number of ints of S which are left fixed by operation f the group G. Then, the number of transity sets is given by

$$N = \frac{1}{n(G)}\sum_{t\in G} I(t) \qquad (2)$$

ere $n(G)$ is the order of the group G. In Slepian's application of this formula, is the set of all 2^{2^n} n-variable Boolean functions, G is the n-dimensional hypercube group of order $n!2^n$, and the transitivity sets are the symmetry-type classes of Boolean functions. The summation in (1) is carried out, not element by element [as in (2)], but in terms of classes of equivalent, *i.e.*, conjugate,[8] operations. All operations of a given conjugacy class C have similar properties. In particular, they have the same value of $I(t)=2^{K(C)}$.

In order to determine the number of self-complementary neutral types in n variables, let S be the space consisting of all (unordered) pairs $(f, \bar{f})$ of complementary functions of n variables. There are obviously 2^{2^n-1} such pairs. The group G of variable transformations is, as before, the hypercube group of order $n!2^n$, consisting of all permutations and/or complementations of variables. One may define two pairs, $(f, \bar{f})$ and $(g, \bar{g})$, in S to be equivalent under G if some operation in G transforms f into either g or $\bar{g}$. This relation between pairs, which is readily seen to be an equivalance relation (*i.e.*, reflexive, symmetric and transitive) then partitions S into classes of equivalent pairs. These classes of pairs will be referred to as *pair symmetry types*. Thus, pair $(f, \bar{f})$ is of the same pair symmetry type as $(g, \bar{g})$ if, and only if, f is of the same (ordinary) symmetry type as either g or $\bar{g}$. For example, the pair consisting of the even parity and odd parity functions forms a pair symmetry type with only one member. Likewise, the pair (f_0, f_1), where $f_0\equiv 0$ and $f_1\equiv 1$, also forms such a class. Most pair classes, however, will contain several distinct pairs.

Let $N_n^{(sc)}$ and $N_n^{(nsc)}$ be the numbers of SC and NSC symmetry types, respectively, of n-variable Boolean functions. The SC types form $N_n^{(sc)}$ pair symmetry types, while the NSC types form $N_n^{(nsc)}/2$ pair symmetry types; hence, the number of pair types is

$$P_n = N_n^{(sc)} + \tfrac{1}{2}N_n^{(nsc)}, \qquad (3)$$

while the number of (ordinary) symmetry types is

$$N_n = N_n^{(sc)} + N_n^{(nsc)}. \qquad (4)$$

Applying formula (2) to this situation, one finds that

$$P_n = \frac{1}{n!2^n}\sum_{t\in G} J(t), \qquad (5)$$

where $J(t)$ is the number of pairs $(f, \bar{f})$ left invariant by operation t of the group G. Now, a pair $(f, \bar{f})$ is left invariant by t if, and only if, either

1) $t(f)=f$, or
2) $t(f)=\bar{f}$.

Let $J_1(t)$ and $J_2(t)$ be the numbers of pairs corresponding to cases 1) and 2), respectively. Thus, $J(t)=J_1(t)+J_2(t)$.

Suppose that the operation t permutes the 2^n vertices of the n cube into $K(C)$ cycles, where C is the operation class to which t belongs. There are $2^{K(C)}$ ways in which these vertices can be labelled with 0's and 1's so that the labelling is constant over each cycle. Each such labeling defines a Boolean function invariant under operation t. Hence, $J_1(t)=2^{K(C)-1}$, since there are half as many pairs as there are functions.

On the other hand, $J_2(t)$ counts the number of pairs $(f, \bar{f})$ such that $t(f)=\bar{f}$. Any such function f corresponds to a labeling of the vertices of the n cube with 0's and 1's alternating as one proceeds around any cycle. Clearly, this is possible only if each cycle is of even length. In that case, there are two possible labelings for each cycle, and the labelings are independent from cycle to cycle. Hence, for all cycles even, there are $2^{K(C)}$ distinct labelings meeting the requirement $t(f)=\bar{f}$; if some cycles are of odd length, there are no such labelings.

Let the operation classes C be separated into two categories—those classes C' whose operations permute the vertices of the n cube into some odd cycles, and those classes C'' whose operations result in all even cycles. We then have

$$J(t) = 2^{K(C')-1}, \quad \text{for } t \text{ in class } C', \text{ and}$$
$$J(t) = 2^{K(C'')}, \quad \text{for } t \text{ in class } C''. \qquad (6)$$

By combining (5) and (6), we obtain

$$P_n = N_n^{(sc)} + \frac{1}{2}N_n^{(nsc)}$$
$$= \frac{1}{n!2^n}\left[\sum_{C'} 2^{K(C')-1}n_{C'} + \sum_{C''} 2^{K(C'')}n_{C''}\right].$$

But, by virtue of (4), we also have

$$N_n = N_n^{(sc)} + N_n^{(nsc)}$$
$$= \frac{1}{n!2^n}\left[\sum_{C'} 2^{K(C')}n_{C'} + \sum_{C''} 2^{K(C'')}n_{C''}\right].$$

It follows that

$$N_n^{(nsc)} = \frac{1}{n!2^n}\sum_{C'} 2^{K(C')}n_{C'} \qquad (7)$$

and

$$N_n^{(sc)} = \frac{1}{n!2^n}\sum_{C''} 2^{K(C'')}n_{C''}. \qquad (8)$$

Thus, the number of self-complementary (hence neutral) symmetry types is given by Slepian's sum (1) restricted to the operation classes C'', while the number of non-self-complementary symmetry types (neutral or not) is given by the sum (1) extended over the remaining operation classes C'. In Table I are shown some values of N_n, $N_n^{(sc)}$, and $N_n^{(neutral)}$ computed with the aid of (7) and (8), and Slepian's formula[3] for the number $N_n^{(neutral)}$ of neutral symmetry types in n variables.

TABLE I
NUMBERS OF SELF-COMPLEMENTARY SYMMETRY TYPES, NEUTRAL SYMMETRY TYPES, AND ALL SYMMETRY TYPES FOR $n=1, 2, 3, 4$ AND 5

n	$N_n^{(sc)}$	$N_n^{(neutral)}$	N_n
1	1	1	3
2	2	2	6
3	6	6	22
4	42	74	402
5	4094	169,112	1,228,158

One notes from Table I that the ratio of $N_n^{(sc)}$ to $N_n^{(neutral)}$ appears to approach zero as n increases. This is strikingly opposite to what one might expect on the basis of the cases $n=1$, 2 and 3 alone, where all the neutral symmetry types are also self-complementary. Thus, for large n it appears that *most* of the neutral types are non-self-com-

[5] M. Karnaugh, "The map method for synthesis combinational logic circuits," *Trans. AIEE (Comn. and Electronics)*, vol. 72, pp. 593–599; November, 53.
[6] G. Pólya, "Sur les types des propositiohs comsées," *J. Symbolic Logic*, vol. 5, pp. 98–103; 1940.
[7] S. W. Golomb, "On the classification of Boolean ctions," IRE TRANS. ON INFORMATION THEORY, IT-5, p. 185; May, 1959.
[8] All operations conjugate to a given operation are of the form ata^{-1} where a is an arbitrary operation of G.

plementary, and that self-complementary types are, in the long run, a rather rare phenomenon. Even for $n=5$, the self-complementary types constitute only about 2.4 per cent of all neutral symmetry types. On the other hand, SC neutral symmetry types certainly exist for all values of n. (The even and odd parity functions have this property, for example.)

It is perhaps pertinent to inquire as to the physical significance of the distinction made above between operations of the group G which belong to classes of category C' and those belonging to category C''. We first point out that any group operation t may be represented in terms of an autonomous sequential network[4] consisting of shift registers and inverters. These components are connected in closed loops, the loops being unconnected to each other, as in Fig. 3, for example. In any loop containing an

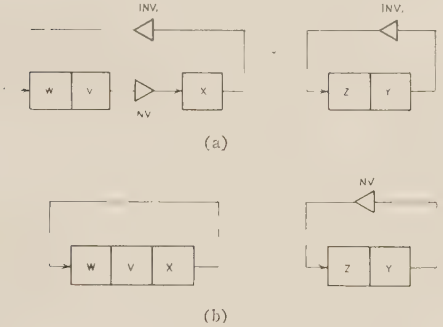


Fig. 3—(a) Network representation of the variable transformation: $v \to w$, $w \to \bar{x}$, $x \to \bar{v}$, $y \to z$, $z \to \bar{y}$. (b) Canonical network representation equivalent to (a).

even number of inverters, these inverters may be removed; and in any loop with an odd number of inverters, all but a single inverter may be removed. The first kind of loop corresponds to what Slepian[3] calls an *e-cycle*. The second kind corresponds to Slepian's *o-cycle*. In replacing an even (or odd) number of inverters by zero (or one) inverter, we change the *specific* operation of the group but not its *class*, C. Thus these networks provide canonical representations of the equivalence classes of the hypercube group. The state diagram of the network then represents the permutation induced by the group operation on the vertices of the n cube.

Now the operation classes of category C'' (all permutation cycles of vertices of even length) may be shown to correspond to operations with some o-cycles. Likewise, the classes C' may be seen to consist solely of operations made up of e-cycles. (These results are implicit in Slepian's work,[3] although no explicit mention is made there of the fact.) In terms of our network model of the situation, the classes C' consist of those operations represented canonically by inverter-free networks, while the classes C'' consist precisely of those operations whose canonical representations contain one or more inverters.

Since the inverter-free nets are particularly simple to analyze, it is most convenient to calculate $N_n^{(nsc)}$ rather than $N_n^{(sc)}$. The determination of the cycle set $[k_1, \cdots, k_{K(C)}]$ for a given network of category C' may be carried out by means of Slepian's Table I, or more directly, by means of the results given by Elspas[9] relating to the cycle set of a circulating shift register. The whole cycle set for all 2^n vertices (*i.e.*, states) is then obtained by the process of cycle set multiplication.[10]

An interesting and as yet unsolved problem is that of obtaining asymptotic expressions of the numbers of SC and NSC neutral symmetry types for large n.

BERNARD ELSPAS
Stanford Res. Inst.
Menlo Park, Calif.

[9] B. Elspas, "The theory of autonomous linear sequential networks," IRE TRANS. ON CIRCUIT THEORY, vol. CT-6, pp. 45–60; March, 1959. See especially p. 60.
[10] *Ibid.*, p. 50.

Correction to "Self-Dual Symmetric Switching Functions with a Certain *a*-Number Constraint"*

There is a typographical error in my letter of the above title.[1] Step 1 of the proof on page 499 should read:

1) By definitions 1 and 2,

$$S_{\{a^j\}}(x_n', \cdots, x_1') \equiv S_{\{\hat{a}\cdot\}}(x_n, \cdots, x_1).$$

The correction consists of putting a roof on the second a_j.

A. A. MULLIN
Dept. of Mathematics
University of Illinois
Urbana, Ill.

* Received by the PGEC, April 20, 1960.
[1] IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 498–499; December, 1959.

Progress Toward Block Diagram Standards*

Many symbols are currently used within the digital computer industry to represent logic functions such as those performed by AND circuits, NOT circuits, flip-flops, and delay-units. Because these symbols conflict with each other, communication between computer groups is not always easy. The acceptance of nationally recognized standards for such symbols would help to reduce the chaos which now exists.

The preparation of national standards for digital computer logical and block diagram symbols was started early in 1956 by the IRE after discussion of the problem by the American Standards Association and others. A special subcommittee is now being set up within the IRE to put the finishing touches on the block diagram symbols for the three logic functions AND, OR, and NOT, as proposed by the computer subcommittee and others. Results are expected shortly.

* Received by the PGEC, March 23, 1960.

The following paragraphs discuss som of the problems that have made it difficult t prepare recognized and accepted standard quickly. The problems can be roughly di vided into two groups: those caused by th fact that there are so many people con cerned, and those caused by the fact tha there are so many symbols to be concerne about.

Many people are concerned with thes symbols because several industries, in addi tion to the digital computer industry, mak extensive use of the logic functions we ar attempting to symbolize. We can mentio just a few such industries: telephone (dialin and switching), railroad (signaling) electri power (distribution control). Many larg segments of industry have widely divergen backgrounds of symbol usage, and show reluctance to accept unfamiliar, hastily con trived symbols. Several methods can be de vised for obtaining standards for such di verse groups. The following four method are worth mentioning:

1) Each big industry could standardiz its own symbols. This method could work i the area of interest of each industry did no include functions being symbolized by an other industry. An attempt was made t use this method within the IRE, but th areas of interest of the computer industr and others overlap so much that this metho proved impractical. It is impractical fo another reason also: there is more conflic of symbols within the computer industr than there is between industries.

2) A single standard could be decided b vote. We have used this method in the IRI for guiding the development of our propose symbols. Voting could be used to decid among several equally acceptable symbol that might evolve after much reasoning an persuasion. But if voting is used to force th selection of a symbol unacceptable to som group, then the method has the disadvantag of selection by dictation.

3) A single standard could be decide by dictation. If the selected symbol is ac ceptable to all, having evolved after reason ing and persuasion, then this method ca work. But a too-hasty and arbitrary choic that is unacceptable to some group coul result in attempts to undermine or circum vent the so-called "standard." In at leas one case, a too-hastily designed symbol ha turned out to be unsatisfactory even to th group who designed it!

4) A single standard could be decide by reason and persuasion. The success of thi method depends on the willingness of ever organization to agree that a given com promise is in the best interests of everybody We have emphasized this method in th IRE. We feel it is the best way of providin an acceptable and recognized standard. Bu it does take considerable time and patienc because there are so many symbols to be con cerned about.

The second complicating factor, the ex istence of many symbols, arises from the ar bitrary nature of symbols and the grea variety of considerations which guide th choice of one for a given application. Th following three properties of symbols ar worth mentioning:

1) A symbol stands for something. Som of its characteristics (such as the number o

nes connected to it) are determined by hat it stands for. There is very little prob- m in the several cases where there happens be a clear-cut way of recognizing such naracteristics.

2) A symbol may be chosen arbitrarily as ng as the few characteristics mentioned bove are satisfied. An appeal to reason can- ot help in choosing among equivalent arbi- rary symbols. This may explain why we an have so many conflicting symbols in the rst place, and why there is so much diffi- ulty in persuading a group to abandon a miliar symbol for one that is unfamiliar ut equally arbitrary.

3) A symbol can be considered a thing itself. Reasonable persuasion, and great nflicts, center around this fact. To men- on a few of the questions to which there re almost as many adamant answers as here are people, we can list these five:

a) Must symbols be easy to draw by hand, by draftsman, or by computer?
b) Must a symbol be mnemonically related to language, logic, or electronics?
c) Should symbols emphasize the similarities or differences between AND, OR, and NOT?
d) Must physical data be kept separate from the logic symbol?
e) Must a symbol reflect present usage (by whom?), or may it be a new creation with no previous history?

Do you know the answers? Caution! There are others who know you are wrong! In other words, people have taken many different strong positions on these and many other questions. Each position is correct within its own context. Since a single solution usually cannot completely satisfy every point of view, it is, in a sense, a compromise. Progress can best be made if each group can see the situation from the points of view of others.

In conclusion, it should be pointed out that in spite of such problems some concrete progress has been made. Our computer subcommittee began by considering most of the digital computer block diagram symbols. As soon as we began to understand the various points of view, we decided to press forward on block diagram symbols for the three logic functions AND, OR, and NOT. It was more of a struggle than we had anticipated. Most of the problems were concerned with symbols in general. Future progress for other functions ought to be more rapid, since most of these general problems have already been considered. A special subcommittee, with representatives from the various industries concerned, is now being set up within the IRE to put the finishing touches on the symbols for these three logic functions. Results are expected shortly. Arrangements are being made for continuing the work on other symbols. This will not necessarily take place in a computer subcommittee because the area of interest in such symbols now appears to be more widespread.

R. P. Mayer
The MITRE Corp.
Lexington, Mass.
(Chairman, IRE Subcommittee 8.9 on Digital Computer Logical and Block Diagram Symbols, and member of Committee 8 on Electronic Computers.)

Contributors

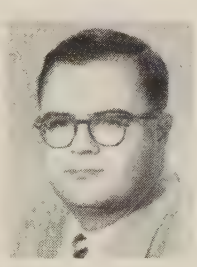
Erik V. Bohn (A'54) was born in Klin- m auf Foehr, Germany, on March 13, 924. He received the Dipl. Math. degree in 1948 and the Dr. Rer. Nat. degree in applied science in 1951 from the University of Goettingen, Germany.

E. V. Bohn

In 1951, he joined the staff of the Research and Consulting Department of the Canadian Marconi Company, Montreal, where he was engaged in the de- gn and development of broad-band an- nna arrays. In 1952, he became a member the staff of the Research and Develop- ent Department of the Canadian Aviation lectronics Company, Montreal, as a pro- ct supervisor. He worked on the design and evelopment of servomechanisms and ana- g computers used in flight and fire control mulation.

In 1954, he joined the staff of the Depart- ent of Electrical Engineering of the Uni- ersity of British Columbia, Vancouver, . C., Can., where he presently holds the osition of Associate Professor.

William F. Caldwell was born in Mays- ville, Ky., on March 18, 1931. He received the B.S.E.E. degree from the Illinois Institute of Technology, Chicago, in 1952 and the M.S.E.E. degree from the University of Kentucky, Lexington, in 1954.



W. F. Caldwell

He then joined the Naval Research Laboratory in Washington, D. C., where he worked in the Transmitter and Receiver Techniques Section. In 1955 he joined the Hughes Research and Development Laboratories at Culver City, Calif., as a member of the Technical Staff. There he worked on experimental test techniques, systems test equipment, and infrared measurements. In 1956 he moved to Tucson, Ariz., to join Hughes' newly formed Tucson Engineering Laboratory, where he had a wide variety of assignments in test equipment design, systems studies, and investigation of operational field test techniques. He was then given charge of a new group organized to complete the final design of the field test equipment for the GAR 3/4 Falcon missile.

In the fall of 1958, Mr. Caldwell left Hughes and enrolled at the University of Arizona, Tucson, to pursue work leading to to the Ph.D. degree in electrical engineering.

❖

Ivan P. V. Carter was born in Dovercourt, Essex, England, on April 6, 1929. He received the B.A. degree in 1950 and the M.A. degree in 1954, both in physics, from Pembroke College, Cambridge, England.

I. P. V. Carter

From 1953 to 1956, he was with Ferranti Ltd., Manchester, England, working first on ferroelectrics and subsequently on magnetic core storage. Since joining the IBM Research Laboratory, Zurich, Switzerland, in 1956, he has worked on magnetic core storage and, more recently, on phosphor measurements.

Mr. Carter is a member of the British Computer Society and of the British Interplanetary Society.

Hsu Chang, for a photograph and biography, please see page 500 of the December, 1959 issue of these TRANSACTIONS.

❖

Hewitt D. Crane (S'47–A'48–M'55–SM'57) was born on April 27, 1927, in Jersey City, N. J. He received the B.S.E.E. degree from Columbia University, New York, N. Y., in 1947, and has done graduate work at Columbia and Princeton Universities. He is currently completing his thesis for the Ph.D. degree at Stanford University, Stanford, Calif.

H. D. CRANE

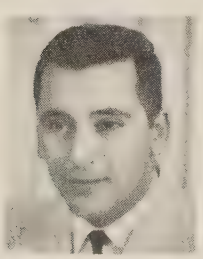
During 1945 and 1946, he served in the U. S. Navy. In 1947–1948, he designed inter-office communication equipment for the David Bogen Company, New York City, N. Y. From 1948–1949, he did development work on facsimile equipment for the Western Union Telegraph Company, New York, N. Y. For the next two and one-half years, his work involved maintenance of the International Business Machines Corporation's S.S.E.C. computer in New York. From 1951 through 1954 at the Institute for Advanced Study, Princeton, N. J., he was concerned with the over-all logic of the IAS computer and was in charge of designing and building the input-output equipment, a graphing device, and a magnetic secondary storage device. For one year at the RCA David Sarnoff Research Center, Princeton, N. J., he worked on magnetic devices, in particular on multi-aperture magnetic devices and on a large-scale magnetic memory.

He joined the staff of Stanford Research Institute, Menlo Park, Calif., in March, 1956. During his first six months at the Institute he was concerned with an analysis of the ERMA logic. More recently, he has initiated work on new circuit components, in particular, multi-aperture magnetic devices. Over twenty inventions in the magnetic devices field have resulted in patent applications.

Mr. Crane is a member of Tau Beta Pi, and the Scientific Research Society of America.

❖

Gerson H. Goldstick (M'57) was born in Trenton, N. J., on January 20, 1933. He received the B.A. degree in mathematics in 1954, the B.S. degree in mechanical engineering in 1955, both from the University of Pennsylvania, Philadelphia, and the M.S.E.E. degree from the University of California at Los Angeles in 1958.



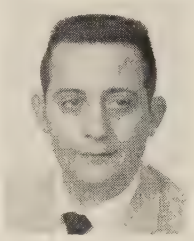
G. H. GOLDSTICK

From 1955 to 1957, he was with the Hughes Aircraft Company, Culver City, Calif., where he worked in digital computer circuit development and reliability engineering. Since 1957, he has been with The National Cash Register Company, Electronics Division, Hawthorne, Calif., where he is at present the head of the circuit analysis and components group. His responsibilities include the design and development of computer circuits. He was a member of the faculty at West Coast University, College of Engineering, Los Angeles, Calif., from 1957 to 1959. He is presently a member of the faculty at UCLA, lecturing on digital computer circuit design and transistor electronics.

Mr. Goldstick is the author of several publications in the areas of computer circuit design and transistor characterizations. He is a member of Phi Beta Kappa, Tau Beta Pi, and Pi Mu Epsilon.

❖

Thomas W. Kampe was born on September 22, 1929, in Los Angeles, Calif. He received the B.A. degree in 1952 and the M.A. degree in 1954, both in mathematics, from the University of Southern California, Los Angeles, where he is currently engaged in postgraduate study.



T. W. KAMPE

He spent two years at West Coast Research, Los Angeles, as a mathematician engaged in analysis and systems study. The following two years he was employed as an IBM punch-card analyst for the Title Insurance and Trust Company, Los Angeles, where he set up punch-card machines for accounting and statistical analysis. He has taught mathematics (from freshman to graduate courses) at the University of Southern California, Pomona and Claremont Colleges, both in Claremont, Calif., and the University of California, Los Angeles.

With Librascope Division, General Precision, Inc., Glendale, Calif., as Senior Mathematician, he conducted mathematical research in target path prediction, filter theory, ballistics, and computer systems analysis. He engaged in logical design of real-time computer systems, both incremental and general purpose, including specialized input/output units. He performed computer programming for both real-time and general scientific computers, including simulation studies, ballistics, curve fitting, linear algebras, statistical methods, and solution of Diophantine equations. He is now Staff Engineer with Librascope's Advanced Projects Group.

Mr. Kampe is a member of the Association for Computing Machinery, American Mathematical Society, Société Mathematic de France, and the Society for Industrial and Applied Mathematics.

❖

William H. Kautz (S'46–A'52–M'57) was born in Seattle, Wash., on February 9, 1924. He received the B.S.E.E. and M.S.E.E. degrees in 1948, the M.S. degree in mathematics in 1949, and the D.Sc. degree in electrical engineering in 1951, all from the Massachusetts Institute of Technology, Cambridge.

W. H. KAUTZ

He was a research assistant in the M.I.T. Research Laboratory of Electronics from 1949 to 1951. In 1951, he joined the staff of Stanford Research Institute, Menlo Park, Calif., where he is currently a Senior Research Engineer in the Computer Techniques Laboratory. He has been engaged in the logical design and testing of digital systems, the development of techniques for the analysis and synthesis of switching networks, and the development of codes and coding systems for computers and communications systems. Several patent assignments have been made.

Dr. Kautz is a member of the American Mathematical Society, Sigma Xi, the Association for Symbolic Logic, the Society for Industrial and Applied Mathematics, and the Association for Computing Machinery.

❖

Granino A. Korn was born in Berlin, Germany, on May 7, 1922. He received the B.A. degree in mathematics and physics from Brown University, Providence, R. I., in 1942, the M.A. degree in physics from Columbia University, New York, N. Y., in 1943, and after service in the U. S. Navy, the Ph.D. degree from Brown University in 1947.

G. A. KORN

From 1946 to 1948, he was a project engineer with the Sperry Gyroscope Co.; from 1948–1949, he was head of the analysis group at Curtiss-Wright Corp.; and, from 1949 to 1955, he was a staff engineer with the Military Operations Analysis Division at Lockheed Aircraft Corp., Burbank, Calif. Since 1957, he has been a professor of electrical engineering at the University of Arizona, Tucson.

His work is chiefly in the field of system design and analog computers, on which subject he has published a number of papers as well as articles in engineering handbooks.

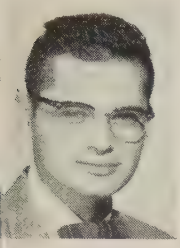
Dr. Korn is a member of Sigma Xi.

❖

Victor R. Latorre (S'55) was born on November 17, 1931, in Brooklyn, N. Y. He received the B.S.E.E. and M.S.E.E. degrees from the University of Arizona, Tucson, in 1956 and 1957, respectively.

During the summer of 1956, he was employed by Newmont Exploration, Ltd., in Jerome, Ariz., where he was responsible for the design of an electromagnetic modeling

stem for the detection of ore deposits. nce 1957, he has been an instructor in the lectrical Engineering Department and a research engineer in the Applied Research Laboratory at the University of Arizona. During this period, he engaged in the development of an automatic meteorological observation system and in studies on tropospheric scatter propagation. He has published papers on transistor circuits nd telemetering systems.



V. R. LATORRE

Mr. Latorre is a member of Tau Beta Pi, igma Pi Sigma, Sigma Xi (Associate), and ie ASEE.

❖

Arthur G. Milnes (SM'58), for a photoraph and biography, please see page 501 of ie December, 1959 issue of these TRANSACIONS.

❖

Thomas H. Mott, Jr., was born in Houson, Tex., on January 24, 1924. He received ie B.A. degree from Rice Institute, Houston, in 1948 and the Ph.D. degree in philosophy from Yale University, New Haven, Conn., in 1956.

T. H. MOTT, JR.

During the war he studied advanced meteorology for a year at New York University, New York, N. Y. under the auspices of the U. S. Air Force. He taught iathematical logic at Yale University before oining the Mathematics Research Departient of Remington Rand Univac at St. aul, Minn., in 1956. Since 1958 he has been mployed at the RCA Laboratories, Princeon, N. J., where he is presently engaged in tudies in problem-solving. He has lectured t the summer engineering sessions of the University of Michigan, Ann Arbor, and is a member of the Electrical Engineering Department of Villanova University, Villanova, Pa., teaching switching theory.

Dr. Mott is a member of Phi Beta Kappa, the Association for Symbolic Logic, and the Association for Computing Machinery.

❖

Gerald R. Peterson (A'52-M'57) was born August 2, 1930, in Oakland, Calif. He received the B.S.E.E. degree from the University of California at Berkeley in 1952 and the M.S.E.E. degree in 1958 from the University of Arizona, Tucson, where he is presently completing work towards the Ph.D. degree in electrical engineering.

G. R. PETERSON

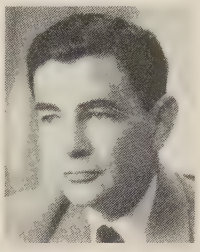
He was a test engineer with General Electric, Schenectady, N. Y., from 1952 to 1953, and, from 1953 to 1955, was a field service engineer with the Aircraft Products Department of General Electric. He entered the U. S. Army in 1955 and was assigned to the Electronic Warfare Department where he worked on evaluation and field testing of surveillance radars. After discharge from the Army in 1956, he joined the faculty at the University of Arizona, where he is presently an instructor in Electrical Engineering.

Mr. Peterson is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.

❖

Hans P. Schlaeppi (A'55) was born in Berne, Switzerland, on April 27, 1926. He received the Dipl. Ing. degree in Electrical engineering from the Eidgenossische Technische Hochschule, Zurich, in 1950.

From 1950 to 1952 he was an assistant at the Institute for Telecommunications of the ETH. From 1952 to 1956 he participated in the construction of the ERMETH computer then being built at the Institute for Applied Mathematics of the ETH; he was responsible for the circuitry and memory development. In 1956, he joined the newly opened IBM Research Laboratory in Adliswil-Zurich where he was in charge of core memory and later of phosphor measurements projects. He is currently doing investigations in machine organization.

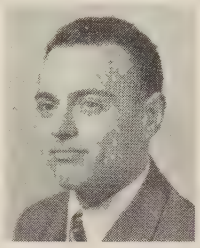


H. P. SCHLAEPPI

Mr. Schlaeppi is a member of the Swiss Physical Society, the Swiss Society for Automation, and the Association for Computing Machinery.

❖

Jack Sklansky (S'52–M'56–SM'59) was born in Brooklyn, N. Y., on November 15, 1928. He received the B.E.E. degree from the College of the City of New York in 1950, the M.S.E.E. degree from Purdue University, Lafayette, Ind., in 1952, and the Doctor of Engineering Science degree from Columbia University, New York, N. Y., in 1955. While at Purdue University, he taught undergraduate electrical engineering on a part-time basis. At Columbia University, he was a Eugene Higgins fellow from 1952 to 1954, and a research assistant at the Electronics Research Laboratory during two summers, and from 1954 to 1955. During previous summers, he was associated with the U. S. Bureau of Reclamation, the U. S. Naval Ordnance Laboratory, and the Bell Telephone Laboratories.



J. SKLANSKY

While at Columbia, he investigated the dynamics of sampled-data control systems, He joined the RCA Laboratories in 1955. where he has conducted research on the logic of fast addition and other digital computer logic, on radar data-processing, on missile dynamics, and on satellite instrumentation. He is presently engaged in research on adaptive system theory.

Dr. Sklansky is a member of the AIEE, the Association for Computing Machinery, Tau Beta Pi, Sigma Xi, and Eta Kappa Nu.

Reviews of Books and Papers in the Computer Field

EDITED BY E. J. MCCLUSKEY, JR.†

Comments and suggestions on this new feature of IRE TRANSACTIONS ON ELECTRONIC COMPUTERS are solicited. Address them to Dr. McCluskey.

All-Transistor Magnetic-Core Memories—B. T. Goda, W. R. Johnston, S. Markowitz, M. Rosenberg, R. Stuart-Williams. [*Commun. and Electronics*, no. 45 (*Trans. AIEE*, vol. 78, pt. 1), pp. 666–673; November, 1959].

In this paper an evaluation is made of linear (word selection) and coincident-current selection modes of magnetic core memory operation, with particular emphasis placed on the use of transistor switching. For large memories (the authors give a word number of 1024 or greater), linear selection is more complex because a separately switched line must be provided for each memory word; but despite this, linear selection has compensating advantages which the authors analyze in considerable detail. Memory interrogation can be done faster since coincidence is not required for this operation, permitting unlimited core drive. The signal-to-noise ratio can be high, even with cores of poor hysteresis-loop squareness, since only the selected core is pulsed. Memory writing also can be faster because partial switching can be utilized.[1] This has the secondary advantage of producing less core heating. The authors emphasize the importance of this by describing the difficult temperature stabilization problems present with coincident selection memories.

Direct core switching with transistors is considered in detail. Here linear selection is again favored, because fewer cores need be driven per selection line, permitting the use of high-frequency, but low-breakdown voltage, switching transistors. A high-speed transistor switch is described which incorporates a very simple current regulator composed of a low-frequency power transistor, in a common base circuit, shunted by a high-frequency diode.

Characteristics of two types of ferrite suitable for linear selection are given. One material is for an 8-μsec memory cycle; the other is for a 2- to 3-μsec cycle. The latter material is of reduced squareness, but this has resulted in cores which switch quickly and exhibit less temperature sensitivity. Evaluation of several current-pulse shapes for reading these cores is included. A slowly-rising triangular waveform was found to be best for maximum one-to-zero signal ratio. An unclear explanation for this is given; in this reviewer's opinion, this result is an expected consequence of the low switching coefficient and poor loop-squareness properties of this promising core material.

The paper includes a comparison of propagation delays for the two modes of memory operation and for magnetic and semiconductor switches. For least delay, the linear selection mode is again claimed to be superior, although little consideration is taken of the large delays which can still be associated with digit-plane inhibition and sensing with this mode. The superiority of the semiconductor compared to the magnetic switch with respect to delay is also described. Here the arguments presented seem to the reviewer to be unchallengeable.

The paper is concluded with a discussion of a difficulty, rather well known to workers in this field, which arises when partially-switched writing is used with low-drive, low-squareness cores. This is the ease with which storage is lost upon application of relatively small information-disturbing pulses of reading polarity. A qualitative explanation is presented in terms of an effective hysteresis-loop "bias" produced by the influence of the adjacent unswitched material upon the switched material in the partially-switched state.

GEORGE R. BRIGGS
RCA Labs.
Princeton, N. J.

† IBM Corporation, Product Dev. Lab., Poughkeepsie, N. Y., until September; then, Dept. of Elec. Engrg., Princeton University, Princeton, N. J.

[1] R. E. McMahon, "Impulse switching of ferrites," *Proc. 1958 E.J.C.C.*, pp. 31–33.

LEM-1, Small Size General Purpose Digital Computer Using Magnetic (Ferrite) Elements—U. A. Machmudov. (*Commun. Assoc. Comp. Mach.*, vol. 2, pp. 3–9; October, 1959.)

The LEM-1 is a 30-kc, single-address, 15-bit binary computer which has a 1024-word operative memory of magnetic cores and a 7167-word permanent "capacitive" memory. The arithmetic unit consists of separate sections for algebraic addition, multiplication, division, and comparison. Switching is performed primarily with magnetic elements and selenium diodes.

The permanent memory contains the program as well as the constants and similar unvarying data. Variable instructions are handled by addressing the operation memory indirectly. The permanent memory uses "the fast permanent capacitative memory cards developed in the Laboratory of Electrical Modeling." One can only speculate as to what this might be.

The operative memory uses two magnetic cores for each bit, and is addressed by means of a coincident switch matrix. The memory is organized into 16 planes, each containing 64 words. Information initially is block-transferred into the operative memory from electromechanically-read paper tape. A magnetic tape unit consisting of 256 blocks, each of 256 words, is available for additional storage. The output consists of an electromechanical printer capable of printing either in octal or decimal. The printing time apparently is one second per word.

The computer contains 3000 magnetic elements (apparently exclusive of the operative memory), 80 vacuum tubes, 500 transistors, and 16,000 selenium diodes. It performs 1200 additions, subtractions, or logical operations, 600 multiplications, and 200 divisions, per second.

Little information is given as to the nature of the capacitance memory, or the novel features in the operative memory. Furthermore, it would be interesting to know of the reliability figures on the 16,000 selenium diodes.

R. C. MINNICK
Stanford Res. Inst.
Menlo Park, Calif.

Handbook of Automation, Computation, and Control—book by E. M. Grabbe, S. Ramo, and D. E. Wooldridge; vol. 2, "Computers and Data Processing." (John Wiley and Sons, Inc., New York, N. Y., 1959. 1033 pages+foreword and preface, $17.50.)

This is the second volume of a three-volume handbook covering the field of automation, computation, and control. The first volume is concerned with mathematical methods, feedback control, information theory, and data transmission. The third volume will be devoted to control systems and components. This second volume provides comprehensive coverage of virtually all aspects of analog and digital computation, including theoretical foundations, components and circuits, system design, and fields of application.

The volume is composed of six sections subdivided into 31 chapters as follows:

A. Computer Terminology
   1. Computer Terminology and Symbols
B. Digital Computer Programming
   2. Programming and Coding
C. The Use of Digital Computers and Data Processors
   3. Data Processing Operations
   4. Quantitative Characteristics of Data Processing Systems
   5. Equipment Description

6. Facility Requirements
7. Design of Business Systems
8. Accounting Applications
9. Inventory and Scheduling Applications
10. Scientific and Engineering Applications
11. Handling of Non-Numerical Information

Design of Digital Computers

12. Digital Computer Fundamentals
13. Techniques for Reliability
14. Components and Basic Circuits
15. Magnetic Core Circuits
16. Transistor Circuits
17. Logical Design
18. Arithmetic and Control Elements
19. Storage
20. Input-Output Equipment for Digital Computers

Design and Application of Analog Computers

21. Analog Computation in Engineering
22. Linear Electronic Computer Elements
23. Nonlinear Electronic Computer Elements
24. Analogs and Duals of Physical Systems
25. Solution of Field Problems
26. Noise and Statistical Techniques
27. Mechanical Computer Elements
28. Digital Techniques in Analog Computation

Unusual Computer Systems

29. Operational Digital Techniques
30. Combined Analog-Digital Computer Systems
31. Simple Turing Type Computers

Section A is primarily a merger of the two glossaries of digital mputer terms issued by the Institute of Radio Engineers in 1956 d the Association for Computing Machinery in 1954. Analog rminology has been overlooked, nor does it appear elsewhere in e book.

Section B comprises over 25 per cent of the book (270 pages), even ough it is included as a single chapter. After an introductory dis- ssion of the nature of programming, flow diagrams, and machine gic, there follows a detailed listing of the complete instruction talogs of nine computers, including the Soviet Strela. An extensive scussion of the objectives, characteristics, and uses of automatic ogramming is found in the succeeding 91 pages. The final 25 pages ver logical programming, microprogramming, programs for equip- ent maintenance, programming with natural language, and an tensive bibliography on programming and related topics.

Section C begins with several excellent chapters in which basic ta processing operations are carefully defined and described, the sential characteristics of data processing systems and equipments e presented, and facility requirements are outlined. The succeeding apters are equally well organized and describe the salient charac- ristics of the various methods for computer solution of a large mber of important problems in five major fields of computer plication.

Section D is of interest primarily to the circuit engineer and logi- l designer. The first two chapters concern themselves with a general scription of digital computer system design, number representa- on, and the use of codes for detecting and correcting errors. The next ree chapters describe the more important switching circuits used in mputers, and discuss techniques for reliable circuit design. The sec- on continues with a comprehensive treatment of principles of logical sign and procedures for the logical design of arithmetic and control its. This is followed by a chapter on magnetic drum and core orage, with brief mention given to acoustic delay lines, electrostatic orage, cryogenic films, and other storage elements. The section con- udes with a long descriptive chapter on input-output equipment, cluding mechanical and electrophotographic printers, devices for unching and reading paper tape, punched card devices; a good dis- ussion of magnetic tape recording and handling; and an excellent rvey of analog-digital converters (both A/D and D/A).

Section E is devoted entirely to analog computers. After a concise ut complete enumeration of the types of analog computers and the asic steps in analog solution of engineering problems, the linear ectronic elements used to mechanize addition, integration, and nstant factors are described. Three fundamental methods for repre- enting complex transfer functions are then detailed. The design of hopper-stabilized proportional amplifiers is discussed in adequate etail. This is followed by an excellent chapter on electronic multi- liers, function generators, and simulation of time delay.

The section continues with a chapter on analogies between mechanical and electrical systems and a table enumerating the corresponding variables in acoustics, heat, and chemistry. This leads to a chapter on analog devices, such as electrolytic tanks and RLC networks, for solving field problems. In the subsequent chapter on noise and statistical techniques, the necessary definitions and mathematical formulas are first presented; devices for generating random noise are then described. A good description of the mechanical differential analyzer is followed by an equally good chapter on the digital differential analyzer (DDA).

Section F, the last section, is a collection of isolated topics correctly labelled "unusual computer systems." The section begins with a description of the hybrid operational digital system in which digital techniques are used in functional units that are laid out in operation form. This is followed by an interesting chapter on a combined analog-digital system in which the analog equipment is used to extend the precision of a digital system at relatively small incremental cost. The final chapter examines questions related to the minimum amount of equipment required to mechanize a digital computer and suggests a measure for comparing different computer designs.

As might be expected in a handbook with more than thirty contributors, the quality of exposition varies. Many authors have simply surveyed their topics without presenting the essential principles. On occasion, the material borders on subjective opinion. Perhaps the most regrettable feature is the unbalance (*e.g.*, 270 pages on programming vs 75 pages on applications of digital computers) and the overlapping (*e.g.*, chapters 5 and 20 on magnetic tape handlers). It is to be hoped that any new edition would be better coordinated.

Although there are many ways in which the exposition and organization could be improved, the fact remains that the volume presents under one cover a comprehensive amount of information of value to designers and users of digital and analog computers. A handbook of the size undertaken here is a monumental task. The editors are to be commended for their efforts in carrying the task through to completion.

Morris Rubinoff
Moore School of Elec. Engrg.
University of Pennsylvania
Philadelphia 4, Pa.

Automation and Computing—book by Andrew D. Booth. (The Macmillan Co., New York, N. Y., 1959. 152 pages+3 index pages.)

Dr. Booth's latest book should appeal to the reader who, although technically inclined, is unwilling to delve into specialized presentations in textbooks or journals in order to survey the developments in the field of modern analog and digital computers, and who wants to study briefly some of the aspects and implications of automation. This book emphasizes principles without resorting to oversimplifications. Several examples, drawn mostly from British industry, are used for illustration. A number of references to more technical discussions is given. For a more general appeal the book should be translated from English into American.

The introductory chapter lists important events in the history of automation and digital and analog computation. Booth's attitude that none of these subjects is of recent origin may disillusion some readers. Some of his conclusions concerning the effects of advances in technology on inflation, population, etc., may provoke argument.

Chapters Two, Three, and Four deal with electronic digital computers. In Chapter Two, the structure of a digital computer is discussed and the basic units are described. Modes of operation and various ways of performing arithmetic are explained by examples. Chapter Three describes some of the electronic building blocks. Chapter Four illustrates the process of programming from mathematical formulation to schematic program and final program. The order code of M.A.C., the machine at Birkbeck College, London, is used. The chapter concludes with some comments on "getting programs right" and the methods which are being used to shorten the debugging process.

Chapter Five briefly describes various analog computing devices which perform analog arithmetic and analog integration, or provide an analog solution of partial differential equations. The solution of hydraulic, aerodynamic, or electrical problems by means of models is also mentioned. The principles involved in electronic analog computers based on the high-gain dc amplifier are presented in some detail.

Chapters Six, Seven, and Eight describe the application of automated procedures to clerical work, to the control of continuous processes, and to machine tools and assembly processes. The possibilities of using automation in an office, not only to perform routine clerical tasks such as calculating income tax deductions (P.A.Y.E. stands for Pay As You Earn), but also to optimize work flow through the factory and to provide up-to-date business information, are mentioned in a general way in Chapter Six. Basic ideas of feedback and stability are discussed in Chapter Seven, together with schematic descriptions of some continuous industrial processes using simple feedback control. Chapter Eight describes several programmed machine tools, with or without position feedback, and some methods of automatic measurement to derive position feedback. The important features of some of the automated assembly processes (Autofab, Mini-Mech, Tinkertoy, etc.) are also listed.

Chapter Nine, "Strategic and Economic Planning," deals with linear programming. As an example, a zero-sum two-person game is discussed. Transportation problems and restriction problems are defined, and two models of economic situations are presented.

The last chapter discusses the applications of (digital) computing machines to such non-numerical tasks as playing games, translating languages, or synthesizing intelligent behavior. The principles of playing Naughts and Crosses (Tic Tac Toe), Nim, Draughts (Checkers), and Chess with a computer are explained, and the difficulties encountered with even a simple game as Naught and Crosses are mentioned. Some comments on the mechanics of language translation and some speculations about thinking or learning by machines conclude the book.

Gernot Metze
Digital Computer Lab.
University of Illinois
Urbana, Ill.

High Speed Computing—book by S. H. Hollingdale. (The Macmillan Co., New York, N. Y., 1959. 259 pages+7 index pages.)

It seems that the British are destined to be three to five years behind the United States both in the computer art and in the writing of books on computers. I suppose the latter follows from the former. At any rate, the book leaves this reviewer cold. Almost nothing is said here that has not been said before in American texts both old and new. Naturally, the illustrative computers, extant machines, are British, which makes them less practical to those who would wish an "available" machine to study.

The preface reveals that, "This book is not intended for the computer specialist, but for the educated general reader; . . . the approach to the subject is that of the user rather than the designer of computing machines; no attempt is made to discuss the more specifically engineering aspects of computer design, construction or maintenance." As such, this book is very similar to:

McCormick, Edward M., "Digital Computer Primer," McGraw-Hill Book Co., Inc., New York, N. Y.; 1959.

But I found the most useful currently available introductory book to be:

Murphy, J. S., "Basics of Digital Computers," John F. Rider, Inc., New York, N. Y., vols. 1–3, 1958.

Written for the service technician, it is profusely illustrated with well-thought-out diagrams, and covers thoroughly computer operation and construction.

For those who are still interested, the contents of Mr. Hollingdale's book are now briefly surveyed by chapter.

1) The organization of the computer, as composed of the arithmetic unit, the store, the control unit, and the I/O units.
2) Number systems.
3) Introduction to programming.
4) Brief History.
5) EDSAC.
6) DEUCE.
7) Storage.
8) Primitive logical design using the British symbols—circles with numbers in them. An OR has a 1 in the circle; an AND has a number equal to the number of inputs. This may be confusing to the novice who may later be exposed to the plethora of American symbology.
9) The discussion of management of a computer installation is not normally found in this kind of book. If the discussion were extended to commercial computers and scientific machines other than EDSAC and DEUCE it might indeed be valuable.
10) A number of examples of applications of scientific computers are interestingly presented and well chosen.
11) The Monte Carlo method is something which many computer people should know about and is briefly presented here.
12) The concept of the control of industrial processes by computers is briefly discussed.
13) The theory of machine translation of languages is explained.

The bibliography of course contains a great deal of British material, but contains little recent American work.

Ivan Flores
Dunlap and Associates, Inc.
Stamford, Conn.

A Comparison of Machine Organizations by Their Performance of the Iterative Solution of Linear Equations—E. J. Gauss. (*J. Assoc. Comp. Mach.*, vol. 6, pp. 476–485; October, 1959.)

The performances of four types of machine organizations are compared by their performance in the solution of simultaneous equations by a particular iterative technique. The organizations considered are single address, four address, single address with index registers, and a "complex organization." The organizations are concluded to be of in creasing efficiency in the order mentioned. With one exception, the organizations are compared quantitatively by calculating the time required for a program written using the order code of an existing computer of each type, but with the assumption of equal add, multiply and access times for the various computers. The "complex" organization considered is one proposed at the University of Illinois and is not described in detail in the paper.

Dean N. Arden
Mass. Inst. Tech.
Cambridge, Mass.

Shift-Register Code for Indexing Applications—M. Nadler and A. Sen Gupta. (*Commun. Assoc. Comp. Mach.*, vol. 2, pp. 40–43; October, 1959.)

There exist at least three different species of codes based on the properties of m sequences. An m sequence of length 2^n-1 may be generated by an appropriate selection of feedback paths of an n stage feedback shift register. Historically, the first of these codes is the one which uses an m sequence and its cyclic permutations as the code words. Another group of codes uses the m sequence to generate parity checks over a set of binary digits.

The crucial property of m sequences used by the authors of this paper is that each 10-bit binary digit (except the all-zero digit) is contained in the sequence once and only once. The type of shift register code discussed uses an m sequence of length $2^{10}-1$, in which to imbed the 64 code words of a standard 10-bit single error correcting code. Since the code words are the addresses of 64 telemetering stations which are being interrogated, this selection leads to certain economies in equipment. The particular example described in the paper was obtained by a straightforward trial and error procedure on a digital computer.

There is no doubt that the authors have described a first-rate solution of a specific engineering problem. Unfortunately, the only result of a general nature provided in this paper is the demonstration once again that m sequences are handy little things to have around.

Norman Abramson
Dept. of Elec. Eng.
Stanford University
Stanford, Calif.

Symbolic Logic and Intelligent Machines—book by E. C. Berkeley. (Reinhold Publishing Corp., New York, N. Y., 1959. 203 pages+5 index pages.)

Mr. Berkeley's book may be characterized as a determined effort to introduce the general reader to the rigors of this field without enforcing boredom on the mathematically sophisticated. It wends a perilous course between a somewhat simple view of intelligence and the sometimes unpalatable intricacies of the algebra of classes. As

uch, it represents a real achievement of great potential value to the verage engineer.

The above paragraph was written in a style somewhat resembling hat used in the book. It, too, could have been considerably improved y the injection of some light-hearted (and enlightening) illustrations t a number of key spots.

The aims and accomplishments of the book must be accorded re- pect, for a highly difficult subject has been presented in terms intel- gible to all. It fills a major gap in our "popular mathematics" books, nd only suffers (as do so many) by comparison with Sylvanus hompson's extraordinary "Calculus Made Easy."

The first five chapters provide an exceptionally clear exposition f the algebra of classes. The correlation with language is expressed imply and effectively, and most of the major results are introduced learly and almost painlessly. It is felt, however, that de-emphasis was too great on the concepts of duality and of canonical forms, which nderlie and can greatly simplify the operating rules; by careful xplanation of their meanings and uses, the quantity and difficulty f the auxiliary rules could have been greatly reduced.

The sixth chapter brings in truth tables and shows their correla- ion with circuits. As a personal reaction, the reviewer found the use f "black boxes" and relay circuits in illustrations rather delightfully uaint for this age of transistors, molecular engineering, and the like.

Chapter 7 serves to introduce the concepts of machine intelli- ence. These are defined in terms of the ability to adapt to new situ- tions. (Something important is lost here in the effort for simplicity, s no real insight is conveyed concerning learning and decision rocesses.) Chapters 8 through 12 provide examples of "intelligent" machines.

Chapters 13 and 14 serve to extend the algebra of classes into the ime domain "of states and events." The over-all results are applied, n Chapter 15, to some simple problems of digital computer program- ning.

The importance of this book derives from the failure of present ooks in the digital computer field to relate the Boolean algebra, vhich they use so freely, to the algebra of classes which underlies t. The necessity for understanding and using the algebra of classes has become more apparent as our logic problems have grown in complexity.

The book contains a rather limited bibliography, but has frequent references to other more specialized (and more difficult) authoritative references in the field.

L. F. Jones
Westinghouse Air Arm
Baltimore, Md.

Algebraic Topological Methods for Contact Network Analysis and Synthesis—C. Saltzer. (*Quart. Appl. Math.*, vol. 17, pp. 173–183; July, 1959.)

Irredundant and Redundant Boolean Branch-Networks—L. Löfgren. (IRE Trans. on Circuit Theory, vol. CT-6, Special Supp., pp. 158–175; May, 1959.)

The Path Matrix and Switching Functions—O. Wing and W. H. Kim. (*J. Franklin Inst.*, vol. 268, pp. 251–269; October, 1959.)

The main concern of the three papers is that of finding linear graphs corresponding to a specified set of paths. The primary purpose of Salzer's paper appears to be expository. The major part of the paper is devoted to a reformulation of familiar properties of linear graphs in algebraic-topological terms. The switching function of the network is expressed as the product of sums of literals, where the literals represent the transmissions of edges. Thus, the edges corresponding to the literals in each sum constitute a cut-set of the graph separating the input vertices. The author derives an (obvious) algorithm for obtaining such an expression from the vertex-edge incidence matrix of the graph. The procedure is reversed for synthesis. However, by over-formalizing, the author overlooks the fact that two of the steps involved are unnecessary. Once the cut-sets are determined from the function and the edges are chosen, the number of vertices in the graph is determined by the rank of the cut-set matrix (and is not a matter of choice). Nor is it necessary to solve any system of linear equations as the author does. The cut-set matrix can be directly reduced to the incidence matrix by elementary operations. The author does not consider the important problem of realizability of the cut-set matrix and the related problem of choosing the edges of the graph. In fact, it may be necessary to augment the cut-set matrix by improper cut-sets (which contain a variable primed and unprimed). Also, he does not prove that the resultant graph realizes the given function. It may, in fact, contain more cut-sets (fewer paths) than the given function unless the cut-set matrix satisfies the condition: The (mod. 2) sum of any odd number of rows corresponds to an admissible cut-set (See R. L. Gould in *Proc. Int. Symp. on the Theory of Switching*, Harvard University, Cambridge, Mass., pt. 1, pp. 244–292; 1957.)

Löfgren on the other hand, is concerned with the important problem of realizability of a given matrix as the circuit matrix of a graph. The main result of the paper is an algorithm for finding the graph corresponding to a given circuit matrix. Like the other known algorithms for this purpose. Löfgren's procedure will yield the graph if one exists. Hence, failure of the procedure is equivalent to unrealizability. However, unlike the other known procedures, Löfgren's scheme is methodical. In essence, Löfgren's procedure is as follows. From the given Boolean function (which is assumed to be realizable with one contact per variable) the circuit matrix is formed, and from it the cut-set matrix, as in S. Okada, *Proc. P.I.B. Symposium on Information Networks*, pp. 267–290; 1954. A graph is drawn for the part of the cut-set matrix which contains at most two 1's per column. The circuit matrix is converted to a fundamental circuit matrix (*i.e.*, containing an identity matrix). The partial graph is now rearranged such that, by adding edges to it, the circuits can be realized one by one. The rearrangements consist of the permissible 2-isomorphic transformations of joining separate parts at a vertex or turning a two-terminal subgraph around, end for end. Although all of the basic ideas are well known and others have used very similar arguments and procedures (see for instance, H. Whitneys' proof that the seven-element matroid corresponds to no graph in *Amer. J. of Math.*, vol. 57, pp. 509–533; 1935.), Löfgren makes a contribution by giving an efficient and organized algorithm. This is offset by unnecessarily complicated notations and proofs which make the paper difficult to read. Löfgren also considers, very briefly, stochastic contact networks. However, the model used is that of information theory—based on the number of contacts that can operate erroneously—rather than the reliability function $h(p)$ of Moore and Shannon. The first part of the paper also contains a review of the elementary aspects of graph theory.

The known matrix-graph synthesis procedures for Boolean functions use an additional edge added between the terminals, to convert paths into circuits. Wing and Kim, on the other hand, prefer to study the paths by themselves without the use of an added edge. (Salzer fails here, too, by adopting neither procedure. Since his synthesis example is the retrieval of a known circuit, he does not get into the usual difficulties.) The first part of the paper contains five theorems about the rank and structure of the path matrix and its relationship to the incidence matrix and switching function. As the authors themselves point out, the rank and structure properties are simply derivable from the known properties of the circuit matrix of a linear graph. The expression for the hindrance function of the network as a Boolean determinant (permanent) is the dual of the procedure due to B. I. Aranovich (*Avt. i Telemekh.*, vol. 10, pp. 437–451; 1949). In part II of the paper, the relationship of the path matrix to the cut-sets separating the input vertices is discussed. A synthesis procedure for realizing the graph from the path matrix is given, based on this relationship. The authors mention the problem of realizability.

It may be of interest to point out that an abstract characterization of cut-set and circuit matrices has been given recently by W. T. Tutte, *Trans. Amer. Math. Soc.*, vol. 90, pp. 527–552; March, 1959.

S. Seshu
University of Toronto
Toronto, Canada

On an Application of Dynamic Programming to the Synthesis of Logical Systems—R. Bellman, J. Holland, and R. Kalaba. (*J. Assoc. Comp. Mach.*, vol. 6, pp. 486–493; October, 1959.)

The authors consider the following problem: an m-bit vector is fed into a series of components each of which transforms the vector into another one of its type. The question to be answered is as follows. With a prescribed set of components, can a network be constructed which converts a given input x into a given output y, and if so, can it be done efficiently?

The method of dynamic programming is used to attack the problem. Given an allowed number N of stages, how close can one get to the desired output? For a metric, the authors choose Hamming distance. However, they neglect to mention that any reasonable metric

would serve equally well and that the choice would depend on the particular problem. If one achieves zero distance in any metric, the problem is solved; *i.e.*, the minimum number of stages has been attained.

Using the metric and the principle of optimality, a functional equation is derived which determines the choice of output to be used at each stage. However, no tractable solution to the functional equation is given. The authors assert that sometimes approximation methods may be used to solve this equation. A simple example is worked out which purports to use the methods and definitions outlined previously. The example is elementary and may be solved by other more straightforward methods.

The paper's main contribution to computer design theory is to show that it is possible to imbed certain minimal design problems in the language of dynamic programming. The authors promise in subsequent papers to treat more realistic problems (and hopefully examples) of the subject with their methods.

Irving S. Reed
Mass. Inst. Tech.
Lincoln Lab.
Lexington, Mass.

On the Construction of Micro-Flowcharts—S. Gorn, P. Z. Ingerman, and J. B. Crozier. (*Commun. Assoc. Comp. Mach.*, vol. 2, pp. 27–31; October, 1959.)

This paper defends the theory that micro-flowcharts, which provide the user of a digital computer with the exact description of the behavior of each instruction of a digital computer under all possible conditions, are useful and should be available to anyone using such a computer. It is unlikely that anyone (with the possible exception of computer manufacturers who do not want to give aid and comfort to competitors) would quarrel with such a premise. The authors also observe that even though micro-flowcharts are usually prepared by the logical designers of a computer, such charts should be written in a language familiar to the users of these charts, who are in most instances programmers. It is true that any manual (or paper) which is to be readily understood by a reader should only assume such knowledge as a reader might normally have. Unfortunately, this article falls somewhat short of this goal because the method for writing micro-flowcharts, which makes up the bulk of the paper, is explained solely through the use of a specific example; the computer for which a sample chart is obtained is a Fieldata computer, and no explanation or description of any kind about the computer, its mode of operation, its order code, etc., is given, so that the reader has to do a certain amount of educated guessing in order to understand the paper well.

The method shows how one can proceed, starting with the logical equations which describe the behavior of the various operations, together with the time sequence into which these equations are embedded, towards a finished micro-flowchart, written in a programmer's language, through a series of intermediate charts and matrices. The early steps of the procedure, as one would expect, are fairly well defined, while the final translation from the designer's language to the programmer's language is a somewhat more ad hoc operation. Some interesting general remarks about this latter process are included in the paper.

The method given illustrates one approach to the problem of writing micro-flowcharts, together with some general thoughts on that problem, and should be of interest to anyone concerned with the construction of micro-flowcharts.

T. A. Dolatta
Princeton University
Princeton, N. J.

Correction

The Editor offers his apologies to William J. Cadden for mislocating him in the December, 1959 issue of these Transactions. Dr. Cadden is with IBM in Poughkeepsie, N. Y., not with Bell Telephone Laboratories, Murray Hill, N. J.

Abstracts of Current Computer Literature

(THROUGH JANUARY, 1960)

These abstracts and the associated subject and author indexes were prepared on a commercial basis under the direction of Dr. Geoffrey Knight, Jr., who also publishes the abstract journal "Semiconductor Electronics." Local volunteer support of this endeavor has been furnished by Messrs. P. R. Bagley and R. P. Mayer of The Mitre Corporation, and F. E. Heart of Lincoln Laboratory, M.I.T.

Additional copies of these abstracts are available from IRE Headquarters, 1 East 79th St., New York 21, N. Y., at $1.00 per copy, or $3.50 for the set of four to be published in 1960.

—*The Editor.*

A-1: EQUIPMENT—THEORETICAL DESIGN

694
Limitations of Microwave Computer by R. C. Stiefel (Sperry-Rand Corp.); PROC. IRE, vol. 48, p. 124 (L); January, 1960.

It is pointed out that as the operating frequencies of digital computers are increased beyond the frequencies presently used, the electromagnetic energy around a conductor is more of a radiating field than one which "clings" to the conductor. Such a radiating field is undesirable in a computer. Therefore, to accomplish high-speed computation it is suggested that paralleling and miniaturization be employed more extensively.

695
A Mathematical Model for Determining the Probabilities of Undetected Errors in Magnetic Tape Systems by M. Schatzoff and W. B. Harding (IBM Corp.); *IBM J. Res. and Dev.*, vol. 1, pp. 177–180; April, 1957.

Mathematical models for evaluating the relative efficiencies of vertical and horizontal redundancy-bit checking in magnetic tape systems are derived. The intuitive concept that two-way checking is greatly superior to vertical checking alone is substantiated, and a method for determining optimum record length is indicated.

A-2: EQUIPMENT—COMPONENTS AND CIRCUITS

696
Electrical Properties of Gold-Doped, Diffused Silicon Computer Diodes by A. E. Bakanowski and J. H. Forster (Bell Telephone Labs., Inc.); *Bell Sys. Tech. J.*, vol. 39, pp. 87–103; January, 1960.

The electrical characteristics of small-area diffused silicon computer diodes in which gold atoms are introduced by solid-state diffusion techniques to control recombination center density and carrier storage time are discussed. Reverse recovery time of about 1 mμsec may be obtained without substantial degradation of other electrical parameters. Comparisons of first-order calculations and experimental results for variations of reverse recovery time, reverse current, and forward current with gold atom density are included.

697
Transistors in Current-Analog Computing, by B. P. Kerfoot (RCA); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 86–93; June, 1956.

Analog computer techniques using transistors with current as the variable are described. Experimental low-power, low-frequency transistor direct-coupled amplifiers to perform summation, scale-change, integration and differentiation are discussed. Factors determining computing accuracy are analyzed and a new technique of error prediction, based on square-wave output, is proposed.

698
The Ferreed—A New Switching Device, by A. Feiner, C. A. Lovell, T. N. Lowry, and P. G. Ridinger (Bell Telephone Labs., Inc.); *Bell Sys. Tech. J.*, vol. 39, pp. 1–30; January, 1960.

An experimental switching device—the ferreed—which has sealed metallic contacts, control times in the microsecond range, coincident selection, memory without holding power, and small size is described. The device may be used as a memory for display, buffer and input-output functions in digital systems. The development of the ferreed is traced from a conceptual model, through realization of a practical model, to possible applications in switching networks. Two methods of coincident control are discussed and three devices related to the conceptual ferreed are briefly described.

699
Solid-State Optoelectronics, by E. E. Loebner (RCA Labs.); *RCA Rev.*, vol. 20, pp. 715–743; December, 1959.

Applications of photoelectric and luminescent phenomena in solids are discussed. Optoelectronic modulators and amplifiers, *i.e.*, devices which have mixed optical and and electrical signal and power access, are described, and the technology of assembling image-transmitting, image-storing, and picture-reproducing panels from optoelectronic elements is reviewed. The functioning of various optoelectronic logic nets and computer components is treated in detail, and synthesis of panel technology and logic circuitry into novel picture-processing panels and computer systems is proposed. The similarity between the organizational structure of such parallel processing systems and that of the neuron network of vertebrate retinas is pointed out.

700
Low Temperature Storage Elements, by E. H. Rhoderick (Services Electronics Res. Lab.); *J. Brit. IRE*, vol. 20, pp. 37–40; January, 1960.

The philosophy underlying the use of low-temperature computer elements is discussed and the cryogenic aspect of the problems briefly reviewed. The most advanced low-temperature storage element at the moment is the Crowe cell, in which a persistent current is set up around an aperture in a thin superconducting film, the direction of the current determining whether a 0 or 1 is stored. The switching time of these elements can be as short as 10 mμsec, and the size is such that between 10^6 and 10^7 can be packed into a cubic foot. The main problem involved in the fabrication of a large memory is that of reproducibility. To exploit the high speed of the Crowe cell, it may be necessary to perform the selection and logical operations in the low-temperature cryostat. Modifications of Buck's original "Cryotron" or avalanche breakdown in a semiconductor could conceivably be used for this purpose.

701
High-Speed Superconductive Switching Element Suitable for Two-Dimensional Fabrication, by V. L. Newhouse and J. W. Bremer (G.E. Co.); *J. Appl. Phys.*, vol. 30, pp. 1458–1459 (L); September, 1959.

The results of experiments on the superconducting-to-normal transition of a tin film due to the magnetic field of current in a transverse much narrower lead film are reported. It is found that by having a large enough ratio between the widths of the two films, an element which can perform the logical and storage functions of a digital computer without the need for intermediate amplification can be constructed. Crossed-strip devices which have a time constant of less than 3 μsec have been constructed. The crossed-strip superconductive element makes practical the construction of non-destructive readout "catalog" memory systems which can increase the performance capabilities of digital computers.

702
Special Solder for Use in Cryogenic Circuits, by A. W. Grobin, Jr. (IBM Corp.); *Rev. Sci. Instr.*, vol. 30, p. 1057; November, 1959.

A solder for attaching leads to an evaporated cryotron is discussed. Silver tabs are fused onto a glass substrate, the cryotron is evaporated on the substrate so as to contact the tabs, and the leads are then soldered to the tabs by a solder which consists of 19 per cent lead, 30 per cent tin, 50 per cent indium, and 1 per cent silver. The solder has a melting point of 130°C, is used with a minimum of nonactivated rosin flux, and is wiped onto the tab with a subminiature soldering iron. The solder contact is strong mechanically, does not fail upon immersion in liquid helium, and appears to be superconducting at 4.2°K. Other solder compositions are listed. These, however, are not as strong mechanically.

703
High-Speed Flip-Flops for the Millimicrosecond Region, by Z. Bay (Natl. Bur. of Standards) and N. T. Grisamore (George Washington Univ.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 121–125; September, 1956.

The design of high-speed flip-flops is approached by dividing the operation into steady-state and switching functions. The steady-state function is controlled by a slave flip-flop and the switching function by the driving circuit. Circuits with conventional components having a resolving time of 10 mμsec are described. Times as low as 2 mμsec have been obtained with special beam deflection tubes. The circuits dissipate considerably less power between switching times than conventional circuits.

704
Current Steering in Magnetic Circuits, by J. A. Rajchman and H. D. Crane (RCA); IRE TRANS. ON ELECTRONIC COMPUTERS vol. EC-6, pp. 21–30; March, 1957.

Magnetic switches in which current from an energizing source is steered through one of many parallel branches by the setting of a core are described. The steering is achieved by core-diode combinations or by transfluxors. Convenient applications are coding and decoding networks, counters, and universal code converters. Economy of associated electronic drivers, current amplitude precision and simple, stable circuitry are among the advantages of the technique.

A-3: EQUIPMENT—SUBSYSTEMS

705
A One-Microsecond Adder Using One-Megacycle Circuitry, by A. Weinberger and J. L. Smith (Natl. Bur. of Standards); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 65–73; June, 1956.

In the addition of two binary numbers, e Boolean expression for any carry digit can be expanded as an explicit function of e input digits of orders k to $k-p+1$ and of e carry digit C_{k-p}. The C_k expressions can simplified by factorization so as to fall thin the gating complexity allowed by the icuitry. A parallel adder capable of adding o 53-bit numbers in 1 μsec is developed using this principle.

6
rvey of Analog Multiplication Schemes, C. M. Edwards (Bendix Aviation Corp.); *Assoc. for Computing Mach.*, vol. 1, pp. –35; January, 1954.

The various methods of analog multiplition that have been proposed are classified to devices depending on a natural physical w, a special function, a variable transmison network, or a modulation process. epresentative members of each type are detribed and the inherent limitations of speed, ndwidth, and stability are analyzed.

7
se of a Diode Ring as a Four-Quadrant ultiplier, by R. H. Wilcox (U. S. Naval es. Lab.); *Rev. Sci. Instr.*, vol. 30, pp. 009–1011; November, 1959.

A diode ring which at low voltage levels rms an extremely simple four-quadrant assive analog multiplier characterized by liability, stability, and wide bandwidth is scussed. The device is based on the inciple that $xy=\frac{1}{4}[(x+y)^2-(x-y)^2]$ and ilizes the current-voltage characteristic of miconductor diodes to accomplish this nction. Operation of the multiplier is nalyzed theoretically and experimental sts are described. These show that with re in selecting diodes and adjusting circuit alues, 1 per cent accuracy is obtainable at put levels up to 150 mv.

8
Time-Division Multiplier, by M. L. ilamand (Société d'Electronique et d'Autoatisme); IRE TRANS. ON ELECTRONIC OMPUTERS, vol. EC-5, pp. 26–34; June, 956.

A time-division analog multiplier which, t a pulse frequency of 2000 cps, has an ccuracy of one part in a thousand, a pass and of 2 cps, an input impedance of one egohm and a very low output, is described. his multiplier has a greater accuracy and ass band, uses less material, and requires ewer adjustments than parabolic diode and ervomultipliers. These results are obtained y the development of a precision electronic witch and by careful attention to compensation of stray capacitances.

9
nalog Multipliers and Squarers Using a ultigrid Modulator, R. L. Sydnor, T. R. 'Meara, and J. Strathman (University of linois); IRE TRANS. ON ELECTRONIC OMPUTERS, vol. EC-5, pp. 82–85; June, 956.

An AM multigrid modulator analog ultiplier with a range of 78 db and only 2 er cent full-scale error is described. This ccuracy is dependent only on the characristics of the tube and not on critical adstment of the operating potentials. The advantages and restrictions of the multiplier, together with a complete range of dynamic performance, are supplied.

710
The Logic of Bidirectional Binary Counters, by M. J. E. Golay (Philco Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 1–4; March, 1957.

The logic of binary counters that can count up or down without waiting for carries completion is examined. Two toggles per stage are required. The possible interconnections between stages are discussed, and an application of the counter to act as an electronic gear to control shaft positions is proposed.

711
High-Speed Shift Registers Using One Core Per Bit, by V. L. Newhouse (RCA) and N. S. Prywes (Sperry-Rand Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 114–120; September, 1956.

Three and two winding per core current-driven one core per bit shift registers capable of operating in the megacycle range are presented and the basic circuitry is analyzed. The state of an intermediate storage capacitor between logical elements is positively controlled by voltage blocking pulses, thereby preventing energy feedback to earlier stages. Applications of the shift register to computer logic operations are described.

712
Six Ways to Use Magnetic Shift Register Elements, by J. Porter (Portronics, Inc.); *Electronics*, vol. 33, pp. 80–81; January 15, 1960.

Six magnetic shift register elements are illustrated, their characteristics are listed, and their operation is briefly described. Both current- and voltage-driven elements are discussed.

713
A New Type of Ferroelectric Shift Register, by J. R. Anderson (National Cash Register Co.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 184–191; December, 1956.

Ferroelectric shift registers consisting of two barium titanate crystals and two or three silicon diodes per stage are described. Operating speeds up to 5 kc with both serial and parallel input and output have been attained. The small size and low power consumption of these devices make them suitable for many computer applications.

714
Memory Matrix Using Ferroelectric Condensers as Bistable Elements, by C. F. Pulvari (Catholic University of America); *J. Assoc. for Computing Mach.*, vol. 2, pp. 169–184; July, 1955.

An electrostatically-operated multicondenser memory matrix using ferroelectric dielectric material is described. Bits are stored in terms of bistable remanent polarization. Individual bits may be addressed either randomly or sequentially by excitation of the corresponding matrix leads. Storage is permanently remanent and has no power requirements. Switching times in the microsecond range are reported.

715
Factors Influencing the Applications of Magnetic Tape Recording to Digital Computers, by D. P. Franklin (EMI Electronics Ltd.); *J. Brit. IRE*, vol. 20, pp. 9–21; January, 1960.

The merit of magnetic tape for storage of digital information and the benefits of two-state operation are briefly discussed. Limitations on the density of recorded information are reviewed to show the extreme precision required in the manufacture of magnetic heads and tape guidance mechanisms. Design features made necessary by high-speed and acceleration requirements are considered with reference to a recently-developed high-performance tape handler.

716
Symposium on Experiences with the Use of Magnetic Tape—2: Magnetic Films on a National-Elliott 405, by P. B. Livesey (Newton, Chambers and Co. Ltd.); *Computer J.*, vol. 2, pp. 120–121; October, 1959.

The construction and operation of magnetic tape units associated with the Ferranti Pegasus and National-Elliott 405 computers are described. In each case, a high degree of reliability and increased speed over the paper tape units which they replace is reported.

717
High-Speed Digital Storage Using Cylindrical Magnetic Films, by G. R. Hoffman, J. A. Turner, and T. Kilburn (University of Manchester); *J. Brit. IRE*, vol. 20, pp. 31–36; January, 1960.

Digital stores consisting of closed magnetic circuits deposited on long glass tubes are described. These promise considerably increased operating speeds compared with present stores and the possibility of producing multielement systems. A system designed to produce 30 tubes with 16 elements per tube in a single evaporation is now operating. Selection modes suitable for an array of this type, which permit greater tolerances than conventional selection systems, have been tested.

718
Magnetic Film File for Computer Storage, by A. St. Johnson [Elliott Brothers (London) Ltd.]; *J. Brit. IRE*, vol. 20, pp. 25–30; January, 1960.

A 35-mm oxide-coated film store in which the pickup head does not contact the oxide is described. The high-quality backing medium provided by the film has resulted in complete freedom from dropouts. Interchangeability between all mechanisms has been achieved with available production heads by using special autostrobing circuits.

719
Impulse Switching of Ferrites, by R. E. McMahon (M.I.T. Lincoln Lab.); *Proc. EJCC*, pp. 31–33; December 3–5, 1958.

The impulse switching technique for increasing the switching speed of ferrite memories is discussed. The amplitude and width of the current drives are adjusted to reduce the size of the core electrically and, therefore, to improve the speed. Switching times as low as 20–50 mμsec are possible with good signal-to-noise ratio. In impulse switching, the write interval switching time

is approximately equal to the read switching time and the power requirements are small.

720
A High-Speed Ferrite Storage System, by C. J. Quartly (Mullard Ltd.); *Electronic Eng.*, vol. 31, pp. 756–758; December, 1959.

A square-loop ferrite storage system which overcomes the limitations on speed imposed by the core material when used in a conventional store with coincident drive selection is described. Each digit is stored as a difference in flux levels in two cores which are only partially switched during the writing operation. Measurements made on a pair of cores show how the output during reading varies with write pulse duration and amplitude and with digit pulse amplitude. The use of this principle with external word selection enables a read/write cycle time of 0.5 μsec to be achieved in a small store.

721
On the Wiring of Two-Dimensional Multiple-Coincidence Magnetic Memories, by N. M. Blachman (Electronic Defence Lab.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 19–21; June, 1956.

The application of Minnick and Ashenhurst's technique of p-fold multiple coincidence magnetic storage in an $n \times n$ array of cores is shown to be equivalent to the problem of finding $(p-2)$ orthogonal Latin squares of order n. By applying reverse currents to the unselected interrogation wires, the disturbance of unselected cores can be reduced to a level lower than that claimed by Minnick and Ashenhurst. If $p=n+1$, this disturbance can be reduced to zero, and $(n+1)$ extra cores can be added at the expense of only one additional interrogating wire. The resulting system is closely allied to the finite projective geometry of order n.

722
On the Optimum Design of Magnetic Drum Store, by D. Dutta Majumdar (Indian Statistical Inst., Calcutta); *J. Inst. Telecommun. Engrs.*, vol. 5, pp. 211–222; September, 1959.

The design of magnetic drum stores is discussed. The factors which must be considered—scanning rate, drum size, packing density, the properties of the coating material and the coating thickness, head design, and switching, are considered. A track switching circuit, which uses a magnetic gating transformer, and which is designed to operate with a high-speed serial drum memory, is briefly described.

723
The Magnetic Drum Store of the "Mercury" Computer, by K. I. Turner and J. E. Thompson (Ferranti Ltd.); *Electronic Engrg.*, vol. 32, pp. 16–21; January, 1960.

A detailed description of the mechanical and electronic design of the magnetic drum store of the high-speed digital computer "Mercury" is given. The main store consists of up to eight magnetic drums, each of which can store 64×3456 bits of information. Each drum is housed with necessary circuits in a self-contained cabinet measuring approximately 5 feet 6 inches $\times$ 3 feet $\times$ 2 feet 6 inches. Electronic track selection on writing and reading permits switching between tracks in less than 20 μsec. Synchronism between the drum and the computer is achieved by means of an electronic system of speed control.

724
A High-Density File Drum as a Computer Store, by L. Knight and M. P. Circuit (Internatl. Computers and Tabulators Ltd.); *J. Brit. IRE*, vol. 20, pp. 41–45; January, 1960.

A large-capacity magnetic drum store having a capacity of ~15,000,000 bits and an average random access time of ~200 msec is described. A packing density of just over 1000 bits per inch has been obtained by floating specially designed heads on a film of oil which automatically maintains a spacing of 0.002 inch between the head and the drum surface. Special considerations led to the use of a copper-nickel-iron alloy for the drum surface. A self-clocked reading system is used to obviate the need for high mechanical stability. The reading circuit also has special features which keep it operating under optimum conditions over a range of signal amplitudes.

725
A Magnetic Disk, Random Access Memory, by A. C. Glover (IBM Brit. Labs.); *J. Brit. IRE*, vol. 20, pp. 22–24; January, 1960.

A large-capacity, random access storage device which uses 50 rotating magnetic disks is described. Total storage capacity is 5×10^5 alphanumeric characters with access time between 0.15 and 0.8 second.

726
Refrigeration of a Superconducting Memory for a Computer, by A. C. Rose-Innes (Services Electronics Res. Lab.); *Brit. J. Appl. Phys.*, vol. 10, pp. 452–454; October, 1959.

The refrigeration required to maintain a memory of superconducting cells is estimated in terms of consumption of liquid helium. Using commercially-available cable, the major source of heat is thermal conduction down the leads to the memory. If a cable of low thermal conductivity is used, a memory of one-million cells should not consume more than two liters of liquid helium per hour. The optimum size for copper electrical leads running directly from room temperature to liquid helium is calculated.

727
Light-Pen Links Computer to Operator, by B. M. Gurley and C. E. Woodward (M.I.T. Lincoln Lab.); *Electronics*, vol. 32, pp. 85–87; November 20, 1959.

A transistorized photoelectric system for reading the cathode ray tube display of the Lincoln Laboratory TX-2 digital computer and several applications of the system are discussed. The light output of a spot on the display is sensed by a germanium photodiode in a pen-shaped holder. The photodiode output is amplified and a previously cleared flip-flop is set to the ONE state. This state indicates to the computer that this point on the display is of interest to the operator. The light pen has several applications. It can, for example, be used to write information into the computer or to control a program in progress. An example of the latter is given.

728
FOSDIC III To Assist in the 1960 Census; *NBS Tech. News Bull.*, vol. 43, pp. 106–107; June, 1959.

FOSDIC III, a machine which can read microfilmed census documents and transcribe the information onto magnetic tape for computer input, is described, and the many advantages of computer input preparation by means of FOSDIC are outlined.

729
A New Diode Function Generator, by H. Amemiya (Showa Denshi, Ltd.) and T. Miura and T. Numakura (Hitachi Central Res. Lab.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 95–100; June, 1957.

Conventional diode function generators that approximate functions by line segments are limited to monotonic functions as the slopes of successive line segments cannot be changed independently. With the generator described, the use of ganged potentiometers makes each segment independent of the rest. Any desired function may be approximated without recourse to oscilloscope inspection. The operating principle and a practical experimental generator are described.

730
Approximation Errors in Diode Function Generators, by N. Ream (Battersea College of Technology); *J. Electronics and Control*, vol. 7, pp. 83–96; July, 1959.

Approximation errors which occur when a piecewise linear function is fitted to a smooth curve in a function generator which utilizes biased diodes are discussed. It is shown how, using either of two obvious criteria for a best fit, the relationship between error and number of segments is obtained from a simple integral to within the accuracy normally required in analog computer applications. The same integral is used to calculate the break-points between segments. Formulas and numerical results for some typical functions are given.

731
Function Generation with Operational Amplifiers, by H. Koerner and G. A. Korn (University of Arizona); *Electronics*, vol. 32, pp. 66–68, 70; November 6, 1959.

Highly-accurate precision limiters for generating nonlinear functions in analog computers are discussed. The high accuracy is achieved by employing high-gain dc amplifiers and voltage feedback. Several circuits which utilize these limiters, including an amplitude comparison circuit, a comparator, a bistable multivibrator, timing circuits, and a resetting circuit, are described. The output voltage of the comparator can be used in computations, unlike that of the conventional comparator which is used mainly for switching or relay-driving. Faster computer operation can be achieved by replacing operational relays with these circuits.

732
An Operational Amplifier with a Differential Input, by V. B. Smolov; *Automation Express*, vol. 1, pp. 20–22, (Complete); February, 1959 [Translation of *Avtomatika i Telemekhanika*, vol. 19, pp. 1145–1149; December, 1958].

The simulation of a number of functions by means of dc operational amplifiers with differential inputs is discussed. The errors involved in the simulation, including both the error inherent in the method and the instrumental error, are considered.

733
The Generation of Squares with the Use of Nonlinear Resistors, by E. Grosswald (Institute for Advanced Study, Princeton University, and University of Pennsylvania); IRE TRANS. ON CIRCUIT THEORY, vol. CT-6, pp. 334–339; December, 1959.

The design of analog computer circuits which generate squares by means of nonlinear resistors, particularly thyrite rods, is discussed. These circuits are usually determined, more or less empirically, by starting from schemes that have given good results in the past and modifying them, or by varying the numerical values of the resistors, used until "best" results are obtained. The concept of best results is clarified and it is shown that, regardless of the complexity of the circuit, the input-output characteristic depends only on two parameters. An explicit method is given by which the numerical values of the resistors used in the thyrite circuit can be computed for the given characteristics of the thyrite rod used in order to achieve desired best results.

A-4: EQUIPMENT—DIGITAL COMPUTERS

734
LEPRECHAUN: A Solid-State Digital Computer, by J. A. Githens and M. J. Gilmartin (Bell Telephone Labs., Inc.); *U. S. Govt., Res. Repts.*, vol. 33, p. 75(A); January 15, 1960. PB 143 648S (Order from LC mi $11.10, ph $69.60).

The TRADIC second feasibility computer, better known as LEPRECHAUN, a high-speed solid-state digital computer capable of solving complex weapons guidance-control problems in real-time, is discussed. The computer was built to demonstrate the feasibility of using direct-coupled transistor logic (DCTL) in a system involving thousands of transistors and to test and evaluate a moderately large magnetic-core memory driven by transistors. It serves as a vehicle for the continuing study and evaluation of the operating characteristics and reliability of transistors in these techniques. LEPRECHAUN was also designed for use in programming and logical design research in the use of digital computers for military real-time control applications. For this purpose, the computer incorporates some novel features which permit extreme flexibility in the logical interconnections. Complete details of the design and construction of the LEPRECHAUN computer are given.

735
The Construction of a Digital-Computing System from a Basic Transistor Circuit, by P. L. Cloot and G. E. Jackson (Metropolitan-Vickers Electrical Co. Ltd.); *Electronic Eng.*, vol. 32, pp. 37–43; January, 1960.

A special-purpose digital computer which utilizes a built-in program for binary-decimal conversion and a basic circuit consisting of one transistor, one capacitor, and three resistors is described. The logical design, circuit design, and the mechanical construction of the machine are discussed. The extremely compact computer uses printed wiring throughout.

A-5: EQUIPMENT—ANALOG COMPUTERS

736
Computing Techniques for the Sampling Parametric Computer, by C. J. Hirsch and F. C. Hallden (Hazeltine Res. Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 108–119; June, 1957.

Analog calculating techniques using exponential discharges to simulate the logarithmic scale of the slide rule are described. Multiplication, division, exponentiation, taking of logarithms, and evaluating nonintegral powers are among the operations performed. Problems may be solved explicitly or implicitly. Accuracies of one or two per cent of full range are obtained. Accuracy can be increased by calibration for specific groups of problems.

737
Respiratory Carbon Dioxide Response Curve Computer, by J. W. Bellville (Memorial Ctr. for Cancer and Allied Diseases) and J. C. Seed (Wellcome Res. Lab.); *Science*, vol. 130, pp. 1079–1083; October 23, 1959.

An analog computer, which can continuously and automatically plot alveolar ventilation-alveolar P_{CO_2} response curves, is discussed. The calculations necessary to obtain alveolar ventilation and alveolar P_{CO_2} are outlined and the operation of the circuitry which performs these computations is described in detail. The input for both calculations is discussed. The output is displayed on an *x-y* plotter. The effects of drugs on respiration can be effectively studied by means of the computer. [See also **The Use of an Analogue Computer for Measurement of Respiratory Depression,** by J. W. Bellville and J. C. Seed; *Trans. N. Y. Acad. Sci.*, ser. II, vol. 22, pp. 34–43; November, 1959.]

738
Repetitive Analog Computer for Analysis of Sums of Distribution Functions, by F. W. Noble, J. E. Hayes, Jr., and M. Eden (Natl. Insts. of Health); PROC. IRE, vol. 47, pp. 1952–1956; November, 1959.

Many experimental procedures yield curves which are sums of distribution functions—for example, electrophoretic, diffusion, and ultracentrifugal patterns, absorption spectra, and curves from countercurrent distribution and from partition chromatography in either liquid or vapor phase. In a given type of curve, each of the component functions is identical to the others in form (for example, Gaussian) but can have very different values of the parameters governing height, width, and position along the abscissa. A computer which determines the parameters for each component by an analysis of the sum curve is described. The computer performs this analysis by synthesizing a number of distribution functions of the desired form, each with adjustable parameters, and by presenting on an oscilloscope the sum of these functions for comparison with the experimental curve being analyzed. A match is made visually by adjustment of the various parameters. When a match has been obtained, the parameters of the component functions are read out, following a switching procedure which presents the individual functions in sequence.

739
Analog Computer for Magnetic Resonance Data Reduction, by R. L. Collins (Phillips Petroleum Co.); *Rev. Sci. Instr.*, vol. 30, p. 492; June, 1959.

An analog computer which can calculate the first moment M_1 of a magnetic resonance absorption curve is described. It is possible to minimize setting errors on the computer. With a good signal-to-noise ratio, M_1 can be calculated with a precision of about 5 per cent.

740
An Analog Computer to Simulate Systems of Coupled Bimolecular Reactions, by E. F. MacNichol, Jr. (Johns Hopkins University); PROC. IRE, vol. 47, pp. 1816–1820; November, 1959.

An analog computer which simulates, as nearly as possible, the flux of material in systems of coupled chemical reactions is discussed. Concentrations of various reactants, intermediates, and products are represented by the potentials at the outputs of electronic integrators. Rates of turnover of materials are represented by charges flowing to and from the integrators. The charges are caused to circulate by means of a "pump" mechanism that transfers charge at a rate proportional to the triple product of three voltages, two of which are derived from integrators and represent the concentrations of reactants. The third represents a rate constant. One voltage controls the frequency of an oscillator, the second controls the duration of a triangular waveform which is triggered by the oscillator, and the third controls its rate of rise. By suitable interconnection of a number of integrators and pumps, a wide variety of reaction schemes can be simulated.

B-1: SYSTEMS—THEORETICAL DESIGN

741
Logical Design of the Digital Computer for the SAGE System, by M. M. Astrahan, B. Housman, J. F. Jacobs, R. P. Mayer, and W. H. Thomas (IBM Corp.); *IBM J. Res.*, and Dev., vol. 1, pp. 76–83; January, 1957.

Special features and performance criteria for the computer used in the SAGE air-defense system are described. Design details for the arithmetic element, high-speed multiply, index registers, input-output, and magnetic-drum buffer are supplied. The system is designed to meet military specifications of speed, capacity, reliability, and flexibility.

742
TRADIC Computer Research (Second Stage), by J. A. Githens and M. J. Gilmartin (Bell Telephone Labs., Inc.); *U. S. Govt. Res. Repts.*, vol. 33, p. 75 (A); January 15, 1960. PB 143 648 (Order from LC mi $11.10, ph $60.60).

A program for the development of solid-state technology for airborne weapons-con-

trol computers is summarized. Among the major items covered are: encoding and decoding techniques; two forms of nondestructive program storage; a wide temperature range magnetic-core memory system; silicon direct-coupled transistor logic; organization of a computer as an aid in trouble location; and programming techniques for the protection of stored constants and error correction in the weapons-control computer.

743
A Photo-Magnetic System for Document and Information Retrieval, by P. James (IBM Corp.); *American Documentation*, vol. 10, pp. 286–295; October, 1959.

A system which can retrieve documents and information from unalterable photographically-stored data and from alterable magnetically-stored data, respectively, is discussed. The operation of the system is illustrated by means of a legal research problem and a prototype design is explained in detail. The stored data can be updated by changes to the magnetic tape and by document-sorting through logic changes in the output program. Several advantages of the system are listed.

744
Micro-Programming, by M. V. Wilkes (Cambridge University); *Proc. EJCC*, pp. 18–20; December 3–5, 1958.

The design of digital-computer sequencing units in which storage for the microprogram is provided by means other than the wiring of the control unit is discussed. Sequencing units which utilize matrices of diodes, a matrix of ferrite cores, or a series of delay lines have been built. The operations of these are described and several advantages of the design technique are pointed out.

745
On Single vs Triple Address Computing Machines, by C. C. Elgot (Naval Ordnance Lab.); *J. Assoc. for Computing Mach.*, vol. 1, pp. 119–123; July, 1954.

The economy of single and triple address machines in coding arithmetical operations is compared. In each case, upper bounds on the number of words necessary to store certain sequences of operations are derived. For certain commonly encountered sequences, it is concluded that the single address organization is more economical.

746
Fingers or Fists, by W. Buchholz (IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 3–11; December, 1959.

A detailed comparison between the binary and decimal number systems in regard to high-speed operations is carried out. The advantages in the binary system of greater memory efficiency and simpler logic in the arithmetic unit are supplemented by great simplification in the control unit in relation to addressing, indexing and instruction modification. Conversely, in applications handling considerable amounts of data, the conversions into binary form may not compensate for the increased speed of binary arithmetic. After careful review, binary addressing and both binary and decimal arithmetic were incorporated in the STRETCH system.

B-2: SYSTEMS—DESCRIPTIONS

747
The Nike Ajax Computer, by W. E. Ingerson (Bell Telephone Labs., Inc.); *Bell Labs. Record*, vol. 38, pp. 26–30; January, 1960.

The design and operation of the analog computer of the Nike Ajax guided-missile system are discussed. The basic techniques employed in the computer are very similar to those employed in the World War II antiaircraft gun directors. The guidance problem of the computer, however, is different since it, unlike the gun director, can continue to control the missile after it is launched. The three phases of the missile guidance problem—the "prelaunch," "early-flight," and "steering" phases—are discussed. A number of features of the missile guidance system are pointed out.

748
Real Time Data Transmission System, by C. R. Scott and W. H. Butler (RCA); IRE Trans. on Communications Systems, vol. CS-7, pp. 201–205; September, 1959.

A real-time data transmission system designed for and installed at the Atlantic Missile Range is described. The system is used to transmit the digital range, azimuth, and elevation coordinates of a missile from remotely located AN/FPS-16 radars into the range safety IBM 704 computer at Cape Canaveral, Florida. The radar data are transmitted at a 10-pps rate and are used in the computation of a predicted impact point ten times a second within the computer. The predicted impact point is displayed on vertical plotting boards for use by the Range Safety Officer.

749
Performance Advances in a Transistorized Computer System: The TRANSAC S-2000, by R. J. Segal, J. L. Maddox, and P. Plano (Philco Corp.); *Proc. EJCC*, pp. 168–174; December 3–5, 1958.

The system organization and the programming assembler-compiler, TAC, of the TRANSAC-S2000 are described. The core memory is directly accessible to the input-output equipment and to the central computer, and a priority system with the tape drives having top priority determines the use of the memory. The machine is equipped with index-registers, is all transistorized, and has a memory read-write cycle of 10 μsec with comparable speeds for arithmetical operations.

750
The IBM 7070 Data Processing System, by R. W. Avery, S. H. Blackford, and J. A. McDonnell (IBM Corp.); *Proc. EJCC*, pp. 165–168; December 3–5, 1958.

The main features of the IBM 7070 high-speed solid-state data processing unit are described. The main storage is magnetic core, backed by magnetic disks and tapes. Word length is ten decimal digits, but arithmetical operation times, being serial, are variable and depend on the length of the data. A feature of the machine is the extreme flexibility of peripheral equipment that may be attached to it.

751
The GE-100 Data Processing System, by R. H. Hagopian, H. L. Herold, J. Levinthal, and J. Weizbaum (G.E. Co.); *Proc. EJCC*, pp. 181–184; December 3–5, 1958.

The input equipment, processing equipment, and output equipment of the GE-100 bank checking account bookkeeping system are described. The most important input equipment is the document handler which reads and sequences the entries. The transistorized central processor operates in a series-parallel mode, using a 4000-word core memory. Memory-cycle time is 32 μsec and a single-address addition requires 64 μsec. Output is provided by a high-speed printer and magnetic tape units provide auxiliary storage.

752
The Formula-Controlled Logical Computer "STANISLAUS," by F. L. Bauer (Johannes Gutenburg University); *Math. Computation*, vol. 14, pp. 64–67; January, 1960.

The design of a special-purpose computer for testing the validity of formulas of the propositional calculus is described. The formulas are written in the parenthesis-free notation of Lucasiewicz, following a rule of Angst that sets up a correspondence between terms of the formula and machine components. The formula may be entered by means of a keyboard. The flashing of a red, yellow, or blue light indicates validity, invalidity, or ill-formation, respectively.

753
SPUD, A Stored-Program Universal Demonstrator for Computer Training, by M. Raspanti (Bell Telephone Labs., Inc.); *Commun. and Electronics*, no. 45 (*Trans. AIEE*, pt 1, vol. 78), pp. 586–594; November, 1959.

SPUD (Stored Program Universal Demonstrator), a stored-program relay machine which is used to teach the programming and operation of stored program information processing systems, is described. The organization of the machine, its "language," and some programs which have been written for it, are discussed.

754
On a Computer for Controlling the Amount of Electrical Energy Which is Fed to An Electric-Arc Furnace for Smelting Steel, by Iu. M. Alyshev, L. N. Fisner, L. I. Shevchenko; *Automation Express*, vol. 2, pp. 19–21 (Excerpts); May, 1959 [Translation of *Avtomatika i Telemekhanika*, vol. 20, pp. 206–210; February, 1959].

The operation of an analog computer which controls the power to an arc furnace is discussed. At each instant of time, the power in the arc is calculated and compared with a specified power value, and the deviation between the two power values is integrated. The output of the computer then compensates for the deviation between the two power values.

B-4: SYSTEMS—TESTING

755
Theoretical Considerations of Routine Maintenance, by E. S. Page (University of Durham); *Computer J.*, vol. 2, pp. 199–204; January, 1960.

A mathematical model of the value of computer operation is proposed. Various functions relating length or error-free run to value are examined, and their consequences are evaluated insofar as they affect the economics of preventive maintenance. It is concluded that maintenance periods should occur just before anticipated long computer runs and that routine maintenance may not be economically justifiable at all for computers whose reliability is near unity.

C-1: AUTOMATA—NATURAL

756

Electronic Control of Some Active Bioelectric Membranes, by J. W. Moore (Natl. Insts. of Health); PROC. IRE, vol. 47, pp. 1869–1880; November, 1959.

The use of special purpose real-time analog computers to measure and control nerve membrane potential or current in a squid axon or a single frog node is discussed. Under current control, the membrane potential has a region of discontinuity and an "action potential" rather similar to that observed in normal impulse propagation. With potential control, the current pattern is a continuous function of the potential, and a negative resistance is found in the region of potential discontinuity for the current-controlled membrane. The membrane's electrical characteristics may therefore be compared with some two-terminal transistor switching circuits.

C-2: AUTOMATA—ARTIFICIAL

757

Two Notes on Machine "Learning", by H. H. Martens (Bell Telephone Labs., Inc.); *Information and Control*, vol. 2, pp. 364–379; December, 1959.

The plan of a program that enables a computer to "learn" to play tic-tac-toe and related 3×3 board games is described. The programmed computer has no built-in knowledge of the game to be played, except for a rule for determining legal moves. It specifically does not "know" what constitutes a win, loss, or draw, but must be informed of the outcome at the end of each play. Experience indicates that a fair competence in tic-tac-toe playing is reached after 30 to 50 plays. Generalizing from this example of a "learning machine," the notation of an L-automaton is introduced, via a formal, behavioristic definition, in an attempt to give an abstract characterization of machine "learning." A solution to the design problem for a general class of L-automata is presented.

758

Penny Matching Machines, by G. M. White (G.E. Res. Labs); *Information and Control*, vol. 2, pp. 349–363; December, 1959.

The design of machines capable of making decisions is discussed by considering the optimum strategy a machine should follow in playing a very simple penny-matching game. At every move, the machine must make a decision to select either a head or a tail. This decision depends on the machine's knowledge of the opponent's characteristics, the opponent's past selections, and the elements of utility that accrue to the machine as a result of each selection. It is desirable that the machine have a minimum amount of analog storage in following this optimum strategy.

759

Current Theory and Practice of Automatic Programming, by S. Gill (Ferranti Ltd.); *Computer J.*, vol. 2, pp. 110–114; October, 1959.

Current techniques in programming are appraised. The subjects considered include automatic programming, recursive definitions, and existing and proposed programming languages. Careful study before implementing a common machine language is emphasized, in view of the large amount of programming time invested in any comprehensive system.

760

A Technique for Handling Macro Instructions, by I. D. Greenwald (RAND Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 21–22; November, 1959.

A technique for the compact definition and description of programmer-defined macro instructions of fixed length is described. Once one macro is defined, it may be a part of a definition of succeeding macros, to any depth desired. Familiarity with the SHARE Assembler for the 704 or 709 is assumed.

761

Multiprogramming STRETCH: Feasibility Considerations by E. F. Codd, E. S. Lowry, E. McDonough, and C. A. Scalzi (IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 13–17; November, 1959.

Minimum requirements for the concurrent execution of several independent programs are discussed. In the STRETCH system, a carefully balanced combination of built-in and programmed logic provides independence of operation, minimum information and maximum control by the programmer, noninterference, automatic supervision and flexible allocation of time and space. Techniques for placing the burden of programmed logic on the compiler and supervisory program rather than on probable programs are described. Machine time accounting and detection of probable program loops are provided for by internal clocks.

762

Input-Output Buffering and FORTRAN, by D. E. Ferguson (University of California); *J. Assoc. for Computing Mach.*, vol. 7, pp. 1–9; January, 1960.

A variable buffering technique that substantially reduces FORTRAN program running times is described. Direct correspondence between physical and logical input and output of data is removed, resulting in much greater flexibility in the use of buffers. A test routine that performs error checks, takes corrective action if necessary, and checks for end-of-tape and end-of-file conditions, monitors the operation of the whole system.

763

Digitized Description Processing, Pt 1. Elementary Considerations, by M. P. Barnett (University of Wisconsin); *U. S. Govt. Res. Repts.*, vol. 32, p. 612(A); November 13, 1959. PB 139 326 (Order from LC mi $3.30, ph $7.80).

Programming systems for the automatic processing by a computer of coded descriptions of mathematical and logical procedures are discussed. It is suggested that by the provision of certain programming programs, which could be developed without too much difficulty, the computer could be used to build up its repertoire of useful programs by the further input of very concise statements of operational procedures.

764

Changing from Analog to Digital Programming by Digital Techniques, by M. L. Stein (University of Minneapolis) and J. Rose (Convair-Astronautics); *J. Assoc. for Computing Mach.*, vol. 7, pp. 10–23; January, 1960.

A compiler program which converts system block diagrams prepared for an analog computer directly into digital computer programs is described. Conditions for implicit outputs are derived, and subroutines for standard blocks such as integrators, adders and multipliers are devised. Outputs from integrators are defined as ultimate dependent variables, and tests for the termination of the substitution procedure are derived. The results may be used to check analog solutions or to obtain greater accuracy for selected parameters.

765

Some Techniques for Dealing with Two-Level Storage, by R. A. Brooker (University of Manchester); *Computer J.*, vol. 2, pp. 189–194; January, 1960.

Techniques developed in the Mercury Autocode for handling the problem associated with two-level storage are described. Particular attention is devoted to the manipulation of matrices stored on a drum. Pseudo-transfer instructions enable the coefficients required for a multiplication to be transferred to fast memory in an efficient manner.

766

The Use of Automatic Programming Techniques for Solving Engineering Problems, by J. T. Carleton, N. Chackan, and T. W. Martin (Westinghouse Electric Corp.); *Commun. and Electronics*, no. 45 (*Trans. AIEE*, vol. 78), pp. 596–601; November, 1959.

Experience with automatic coding systems on four large-scale digital computers (one IBM 705 and three IBM 704's) at the Westinghouse Electric Corporation is summarized.

767

Frameworks I and II: Automatic Programming Systems for UNIVACS I and II, by A. Shapiro (David Taylor Model Basin); *U. S. Govt. Res. Repts.*, vol. 32, pp. 611–612(A); November 13, 1959. PB 142 518 (Order from LC mi $3.60, ph $9.30).

The capabilities and limitations of Frameworks I and II, automatic programming systems designed for UNIVAC, are described and full directions on how to use these systems are given.

D-1: PROGRAMS—AUTOMATIC PROGRAMMING, DIGITAL COMPUTERS

768
OMNIFORM 1—A General-Purpose Program for High-Speed Computers; *NBS, Tech. News Bull.*, vol. 44, pp. 8–9; January, 1960.

OMNIFORM I, an interpretive program for instructing a machine to perform a number of specialized calculations required in many fields of physics, chemistry, and engineering, is discussed. At present, the program has 16 functions (power series, natural and common logarithms; and exponential, trigonometric, and hyperbolic functions) built in and, in the future, it will include Bessel functions and Legendre, Laguerre, Hermite, and Tchebycheff polynomials. The program was assembled using the FORTRAN programming system with subprograms from the FORTRAN library of routines. OMNIFORM is primarily a table generator and can produce a table of up to 50 entries (50 different functions) for each of 100 arguments.

769
A Function Interpretive Scheme for Pegasus, by J. S. Hornsby (Hawker Aircraft Ltd.); *Computer J.*, vol. 2, pp. 174–180; January, 1960.

An automatic programming system designed for the Ferranti Pegasus machine, whereby complete mathematical formulas may be evaluated without breaking them down into their component arithmetical operations, is described. The machine itself programs the formulas and processes the data without the need for detailed instructions. The scheme is similar in concept to FORTRAN, though somewhat less flexible.

770
RUNCIBLE, Algebraic Translation on a Limited Computer, by D. E. Knuth (Case Inst. Tech.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 18–21; November, 1959.

The main features of RUNCIBLE I, an algebraic compiler developed at Case Institute of Technology for use on IBM 650 installations, are outlined. The system is compatible with the information theory language and completes the translation into machine language in one pass. The techniques employed on RUNCIBLE are highly machine-oriented in order to keep the compiler efficient and yet retain its usefulness for a machine with limited internal capacity.

D-2: PROGRAMS—APPLICATIONS, DIGITAL COMPUTERS

771
Data Processing and Operations Analysis on a Scientific Computer; *NBS Tech. News Bull.*, vol. 44, pp. 8–9; January, 1960.

Data processing and operations analysis problems which have been, or are presently being, solved on the National Bureau of Standards' IBM 704 computer are discussed. Among these are a war-game study called Autotag; the design of computer programs for analyzing human heartbeats; an analysis of Government-subsidized low-rent housing in which letters to regional and project authorities are automatically printed; the use of a computer to work out design details for a new computer; and the mechanical translation of Russian.

772
Data Processing and Information Handling, by R. H. Gregory and M. Trust (M.I.T.); *Proc. EJCC*, pp. 65–71; December 3–5, 1958.

The mechanical selection of "information," *i.e.*, the facts necessary for management decision-making, from "data," *i.e.*, all the available facts, is discussed. Factors for selecting items, or exceptions, for management action are considered, and an exception-processing program for an IBM 704 is described.

773
Time-Series Analysis, by J. Shiskin (Bureau of the Census); *Univac Rev.*, pp. 4–8; Fall, 1959.

The analysis of time series for seasonal, cyclical, and irregular components by such means as periodic moving averages is described. The derivation of changing trends from the cyclical components and the use for management decision-making of information extracted from the analysis is commented on.

774
Automation Trends in the Banking Industry, by F. M. Miller (Natl. Assoc. of Bank Auditors and Comptrollers); IRE TRANS. ON INDUSTRIAL ELECTRONICS, vol. PGIE-10, pp. 3–7; July, 1959.

Data-processing problems in the banking industry are reviewed. These problems are now being investigated on an industry-wide basis. It is hoped that a uniform accounting system will be adopted by the nation's 14,500 banks. Such a uniform accounting system will permit manufacturers to design and build data-processing machines which can be used in all banks. It is suggested that computers can be used to simulate bank operations and that centralized data-processing centers be established for the use of several banks. Such centers will permit funds to be transferred without the need of writing a check.

775
A Business Application of a Digital Computer, by A. G. Wright (The Imperial Tobacco Co. of Great Britain and Ireland Ltd.); *Computer J.*, vol. 2, pp. 103–104; October, 1959.

The application of a Leo II Computer to invoice and inventory control for a large tobacco company is described. Numerous decisions as to package specifications and routing odd lots and customer credit are handled by the program. High criteria of reliability and maintenance are imposed on the system.

776
Early Experience with an E.D.P. System, by T. C. Hickman (UHAS Electronic Computer Service, London); *Computer J.*, vol. 2, pp. 152–163; January, 1960.

The experience of setting up an electronic data-processing system for a large manufacturing and distributing company is described. Problems of change-over, choice of computer, recruitment, and maintenance are discussed. Applications of the system include payroll, order-processing, auditing, inventory, and statistics.

777
A Queue Network Simulator for the IBM 650 and Burroughs 220, by R. W. Conway, B. M. Johnson, and W. L. Maxwell (Cornell Univ.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 20–23; December, 1959.

Two programs, CORE I and CORE II, written specifically for the IBM 650 and Burroughs 220, respectively, to simulate "job-shop" operations are described. Up to 100 jobs with a maximum number of seven operations per job can be accommodated. Nine machine groups with a maximum of ten machines per group may be considered. A wide variety of priority rules depending on order of arrival, value of job, due time, release time, and estimate of working time are simulated. Applications include the comparison of different priority rules and the study of bottlenecks, job routing, and machine specification flexibility.

778
Experience in Using a DEUCE Computer for the Family Expenditure Survey, by P. Redfern (Central Statistics Office, London); *Computer J.*, vol. 2, pp. 164–169; January, 1960.

The application of a digital computer for the study of family expenditure is described. Choice of computer, input-output media, program and computer operation, and costs are discussed.

779
Forecasting Election Results, by D. Milledge and M. J. Mills (Ferranti Ltd.); *Computer J.*, vol. 2, pp. 195–198; January, 1960.

The forecasting of election results by using estimates of percentage swings based on previous elections, early results, and Gallup Polls is described. Adjustments corresponding to three-cornered elections and incumbent candidates are discussed. Particular attention is paid to checking the accuracy of incoming data. Results forecast for the recent British General Election were in close agreement with the final figures.

780
The Use of Computers for Economic Planning in the Petroleum Chemical Industry, by G. S. Galer (Shell Chemical Co. Ltd.); *Computer J.*, vol. 2, pp. 145–149; October, 1959.

The use of computers to solve optimum-mix problems in the petroleum industry is described. As all the variables may be linearized to a first approximation, linear programming using the simplex method of solution is the tool most conveniently employed. Optimum profit solutions frequently result in unsaleable surpluses of certain products. The cost of various policy decisions under such circumstances may be evaluated.

781
The Introduction and Establishment of a System of Computer Production Control in a Light Engineering Factory, by J. F. A. Bryen (Internatl. Computers and Tabulators, Ltd.); *Computer J.*, vol. 2, pp. 115–118; October, 1959.

The application of computer methods to

e problem of production control in a light gineering factory is described. The bene- s include a reduction of items in short pply, increased productivity, better stock llance, and provision of simulation tech- ques for management decision making.

2

roblems of Local Authorities in Data rocessing, by C. W. Mallinson; *Computer* , vol. 2, pp. 105-107; October, 1959.

Questions of volume data, reliable trans- ission, transition from the old system to e computer system, and recruitment of ained staff, insofar as they apply to local thorities, are discussed. Sharing of com- ter facilities by several authorities (where ppropriate), planning for a carefully inte- ated system, and using a building-block pproach, are recommended. Reliability and pid input-output are important features computers in this application.

3

pplication of a Combination of Analog nd Digital Computers to Electron Tra- ctory Tracing, by J. Vine (Siemens Edison wan Ltd.); *Computer J.*, vol. 2, pp. 134–144; ctober, 1959.

Two methods for tracing the paths of harged particles through electrostatic fields re described. Each uses a digital computer conjunction with a resistance network. he first, an application of the ray-tracing ethod, is suitable for the computation of ns properties, while the second is of more eneral scope. Analytically-solvable test ases indicate a high degree of accuracy by oth methods.

4

ormulation of Transparent Colors with a igital Computer, by F. W. Billmeyer Jr., K. Beasley, and J. A. Sheldon (E. I. du ont de Nemours and Co.); *J. Opt. Soc. mer.*, vol. 50, pp. 70–72; January, 1960.

The calculation on a digital computer of he concentrations required to produce a iven transparent color by mixing soluble yes is discussed. The computation is based n Beer's law calculations at 65 wavelengths cross the visible spectrum. Running time or the color-matching programs, which have een run on several computers including the NIVAC and the IBM 650, depends on the achine and details of the program, but is ften less than five minutes per formulation. he color to be formulated is specified in erms of CIE tristimulus values derived from nstrumental measurement. The computer ormulation technique was tested by making p to computed formulas a series of mixtures f dyes in a solvent and in an acrylic resin. he correspondence was good between the easured colors of the mixtures and those or which the formulations were calculated.

5

alculation of Order Parameters in a Binary lloy by the Monte Carlo Method, by L. D. osdick (University of Illinois); *Phys. Rev.*, ol. 116, pp. 565–573; November 1, 1959.

The use of a Monte Carlo sampling heme, similar to that used by Metropolis, Vood, and others, in equations of state omputations for gases, to investigate order- isorder phenomena in a face-centered cubic $_3B$ alloy is discussed. The calculations were performed on an IBM 704 computer using a program known as Monte Crysto. The model of the alloy assumes that the structure of the lattice is fixed and that interactions exist between first neighbors and second neighbors only. In most of the calculations, detailed consideration is given to an array consisting of five unit cells on an edge (five-hundred sites) with periodic boundary conditions. The long-range order and short-range order for first and second neighbors have been computed above and below the critical temperature. Using the energy parameter $v_n=[(V_{AA^{rn}})+V_{BB^{rn}}/2]-V_{AB^{rn}}$ for nth neighbors, it is found that $v_2/v_1=-0.25$ and $v_1=816$ cal/mole gives the best agreement with experiments on Cu_3Au. The critical temperature appears to vary linearly with the ratio v_2/v_1.

786

System Synthesis with the Aid of Digital Computers, by J. B. Dennis, R. F. Nease, and R. M. Saunders (M.I.T.); *Commun. and Electronics*, no. 45 (*Trans. AIEE*, vol. 78) pp. 512–515; November, 1959.

The conventional procedure for designing systems with many physical parameters is discussed and the translation of the procedure into digital computer operations is considered. Although it is possible to code all of the routine design calculations for a computer, it is impossible to program the insight of the design engineer. It is shown that the determination of the physical parameters for the optimum design of a system is generally equivalent to the minimization of a function subject to constraints. The application of the method of steepest descent to minimization problems is discussed and a method for extending its use to minimization with constraint is presented.

787

The Application of Digital Computers to the Reduction and Analysis of High-Dispersion Molecular Spectra, by J. G. Phillips (University of California); *Astrophys. J.*, vol. 130, pp. 308–323; July, 1959.

Programs which have been developed for the IBM 701 digital computer, for the calculation of vacuum wave numbers from line measurements, and for the identification of the branches into which the lines of individual bands may be divided, are discussed. Tests on representative spectra show that great savings of time and labor may be achieved, especially if the programs are used in conjunction with digitizing equipment on the measuring machines to punch the measurements directly onto IBM cards.

788

Some Remarks about Crystallographic Calculations on High-Speed Digital Computers, by M. G. Rossmann, R. A. Jacobson, *et al.* (University of Minnesota); *U. S. Govt. Res. Repts.*, vol. 32, p. 633(A); November 13, 1959. PB 142 460 (Order from LC mi $2.70, ph $4.80).

A general least-squares program which deals with a method of introducing the space group symmetry into both positional and thermal parameters of each atom during crystallographic calculations on high-speed digital computers is discussed. The choice of a damping factor is considered in relation to convergence. An example in which the least-squares method was able to show up two incorrectly-placed atoms in an otherwise-correct structure is given. A one-letter code which represents electron density is described as a way of overcoming the output difficulty for a general Fourier summation program.

789

Great Circle Calculations by Digital Computer, by B. E. Trotter (Naval Res. Labs.) *U. S. Govt. Res. Repts.*, vol. 32, p. 612(A); November 13, 1959. PB 140 185 (Order from LC mi $2.40, ph $3.30).

A 12-track program (excluding standard subroutines), which performs great circle calculations required in connection with direction finding, radio propagation, and navigation problems, is discussed. The program which was written for the LGP-30, a small, general-purpose digital computer, is characterized by simplified input and concise, unambiguous output.

790

Electronic Computers in Surveying, by A. J. McNair (Cornell University); *Trans. Am. Soc. Civil Engrs.*, vol. 124, pp. 252–257; 1959.

The use of electronic digital computers to solve routine and complex surveying problems and to teach surveying is discussed. Computers can relieve the civil engineer of routine, repetitious, and time-consuming computations and permit him to solve problems—for example—those associated with photogrammetry, which would otherwise be too complex to attempt. The student stands to gain much from the use of a computer in a surveying course. The instructor also gains flexibility in assigning problems since he can assign the same problem, but with different values, to each student. The instructor can solve each problem rapidly on a computer.

791

Exponentially Mapped Past Statistical Variables, by J. Otteman (University of Michigan); *U. S. Govt. Res. Repts.*, vol. 32, p. 613(A); November 13, 1959. PB 142 781 (Order from LC mi $2.40, ph $3.30).

The analysis of battlefield-surveillance data by an analog or a digital computer is discussed. In the constantly changing picture of battlefield conditions, interest is focused on recent events. Exponentially mapped past (emp) statistical variables are quantities relating to a set of observations computed in such a way that the recent values of the observations contribute more strongly than the values observed in the more distant past. The relative weighting is a geometrical ratio in the case of discrete (naturally discrete or sampled) data and an exponential function in the case of continuously observed functions. Definitions of some emp variables are introduced and certain simple relationships are discussed. The distinct computation advantages of the emp variables are pointed out.

792

Use of Indicator Concentration Curves in Computation of Mean Rate of Flow and Volume of Blood Contained Within a Segment of the Vascular System, by H. D. Green, A. B. Denison, Jr., C. E. Rapela, and G. Lin (Wake Forest College); IRE TRANS. ON

MEDICAL ELECTRONICS, vol. ME-6, pp. 277–282; December, 1959.

A mathematical analysis is performed to show that indicator concentration curves measured at a point downstream from the point of injection of an indicator can be used to calculate the mean flow of blood at either the upstream or the downstream point and that a curve obtained by the integration of the quantity (indicator concentration $\times$time) for all slugs of fluid can be used to determine the mean transit time of the fluid and, therefore, the volume of the fluid between the two points. The integration can be performed by a computer, provided that a linear relationship can be obtained between recorder deflection and indicator concentration.

793

An Experiment in Musical Composition, by F. P. Brooks, Jr. (IBM Corp.), A. L. Hopkins, P. G. Neumann, and V. W. Wright (Harvard University); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 175–182; September, 1957.

By analyzing the probabilities of elements, element pairs (digrams), trigrams, etc., in a sample of melodies, a table of probabilities is derived. With the aid of the table, original melodies are synthesized by a random process. Experimental results presented include comparative statistics of the successful syntheses using eight orders of analysis. Possible applications of the analysis-synthesis technique employed to a wider class of problems are discussed.

794

Time-Sharing on the National-Elliott 802, by R. L. Cook (Elliott Brothers (London) Ltd.); *Computer J.*, vol. 2, pp. 185–188; January, 1960.

Programming techniques that handle real-time programs in the process-control field on the National-Elliott 802 computer are described. A time-sharing system of program-interruption, which is automatic but manually alterable, is provided. A priority system enables most efficient use to be made of the computer and provides for convenient interleaving of input and output. In applications where the machine is not fully occupied, the time-sharing principle enables independent programs to be run simultaneously with the main program.

795

Information Storage and Retrieval Using a Large Scale Random Access Memory, by J. J. Nolan (IBM Corp.); *American Documentation*, vol. 10, pp. 27–35; January, 1959.

The use of the IBM 305 RAMAC as an information retrieval tool is described. Random entry to such a large capacity memory and the associated programmable features may make coordinate searching of large collections practical since such problems as the recognition of specific-generic relationships and false association between search terms can be overcome.

796

A System for the Correlation of Physical Properties and Structural Characteristics of Chemical Compounds with their Commercial Uses, by R. A. Carpenter, C. C. Bolze, and L. D. Findley (Midwest Res. Inst.); *American Documentation*, vol. 10, pp. 138–143; April, 1959.

Computer techniques for scanning the estimated three-quarters of a million known chemical compounds to determine those which meet certain criteria are discussed. The criteria may be based on such properties as melting and boiling points, refractive index, molecular weight, and empirical and structural formulas. Codes to represent these data for storage on magnetic tape and subsequent comparison using an IBM 704 computer are described.

797

The Use of Electronic Computers in Medical Data Processing: Aids in Diagnosis, Current Information Retrieval, and Medical Record Keeping, by R. S. Ledley (Natl. Acad. Sci.—Natl. Res. Council and George Washington University) and L. B. Lusted (University of Rochester School of Medicine and Dentistry); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 31–47; January, 1960.

Some of the potential advantages of computer aids to medical data processing are discussed. These are: making available to the physician quantitative methods in areas relating to data analysis and differential diagnosis; assisting in the evaluation of the best alternative courses of action during stages of the diagnostic testing processes; making easily available to the physician reference to the most current information about new preventative measures and diagnostic and therapeutic techniques; and periodic recording and evaluating of individual physiologic norms for the more sensitive determination of an individual's health trend relative to disease prevention (22 references). [See, also, **The Use of Electronic Computers to Aid in Medical Diagnosis,** by R. S. Ledley and L. B. Lusted; PROC. IRE, vol. 47, pp. 1970–1977; November, 1959.]

798

Display of Chemical Structure Formulas as Digital Computer Output, by A. Opler and N. Baird (Dow Chemical Co.); *American Documentation*, vol. 10, pp. 59–63; January, 1959.

A general method for displaying figures on the cathode ray output of an IBM 704 computer is discussed. The method is described in particular for the display of organic chemical structural formulas, but can also be used for displaying electronic circuit diagrams, computer flow-charts, highway maps, and the like. The preparation of the structural formula depiction code is described and the processing necessary to prepare a magnetic tape which contains the code and which is compatible with the computer is outlined.

799

A First Approach to Patent Searching Procedures on Standard's Electronic Automatic Computer (SEAC), by H. Pfeffer and H. R. Koller (U. S. Patent Off.) and E. C. Marden (Natl. Bur. of Standards); *American Documentation*, vol. 10, pp. 20–26; January, 1959.

The four routines of the HAYSTAQ program for conducting technical literature searches on the SEAC computer are described. The program has been developed for use by the U. S. Patent Office and is presently restricted to the field of chemistry. The HAYSTAQ program can handle diverse relationships among chemical compounds *per se* and among their structural elements and can examine documents with respect to negative teachings. The applicability of HAYSTAQ to large-scale searching operations is limited by the necessity of serial inspection of documents. It is pointed out that machines which can search a document file in a parallel manner are necessary for such searching.

800

Automatic Character Recognition, by D. A. Young (Natl. Cash Register Co. Ltd.); *Electronic Engrg.*, vol. 32, pp. 2–10; January, 1960.

Following an introductory survey of some of the principal automatic character-recognition systems that are either in production or being investigated, the fundamental semantic features and limits of character patterns are discussed. It is demonstrated that by an analysis of these limiting features, it is possible to formulate definitions of character patterns that are particularly amenable to translation in terms of discriminatory logic.

801

Machine Translation of Russian; *NBS Tech. News Bull.*, vol. 43, pp. 101–102; June, 1959.

A program which employs an iterative scheme to perform mechanical translation of Russian to English is described. When this scheme fails, a word-to-word translation is printed out with an indication of where the scheme failed. A feature of the scheme is its ability to predict associations among words. The iterative translations are intelligible although they are not perfect English.

802

Results Obtained from a Vowel Recognition Computer Program, by J. W. Forgie and C. D. Forgie (M.I.T.); *J. Acoust. Soc. Amer.*, vol. 31, pp. 1480–1489; November, 1959.

A program developed for use on the Whirlwind 1 computer to recognize ten English vowels in isolated words of the form $|b|$-vowel-$|t|$ is discussed. The input to the computer was real-time spectral data. The program first determined the rough location of the first two formants. The remaining confusions were resolved by 1) finer determination of the F-1 and F-2 locations by the use of slope and/or valley information, 2) the use of pitch information, and 3) in certain cases, the determination of the position of F-3. The over-all score for 21 subjects (11 male and 10 female) was 88 per cent. By the use of duration information, the score was raised to 93 per cent.

803

The Use of the IBM 704 in the Simulation of Speech Recognition Systems, by G. L. Shultz (IBM Corp.); *U. S. Govt. Res. Repts.*, vol. 32, p. 612(A); November 13, 1959. PB 142 585 (Order from LC mi. $2.70, ph $4.80).

The first step in mechanical speech recognition involves the analysis of a large number of speech sounds to determine the characteristics by which these sounds may best be discriminated. Equipment required to facilitate editing samples of sounds for analysis and to convert these sounds to digital form suitable as computer inputs is described. A system of programs is presented, and the feasibility of the computer as a research tool is illustrated.

D-3: PROGRAMS—TECHNIQUES, DIGITAL COMPUTERS

804

The Generation of Pseudo-Random Numbers on Electronic Digital Computers, by J. R. Edmonds (University of London); *Computer J.*, vol. 2, pp. 181–185; January, 1960.

Different methods of generating sequences of pseudo-random numbers, including the middle squares, Fibonacci sequence, and various modular reduction methods, are discussed. The statistical tests of randomness that can be applied to such series are considered, and pseudo-random number routines that have been written for the Ferranti Pegasus and Mercury computers are described.

805

A New Pseudo-Random Number Generator, by A. Rotenberg (New York University); *J. Assoc. for Computing Mach.*, vol. 7, pp. 75–77; January, 1960.

An additive congruential method of generating pseudo-random numbers based on the sequence $x_{i+1}=(2^a+1)xi+c(\text{mod } 2^{35})$ is described. It is shown that the sequence generates the full period of 2^{35} numbers for $a=2$ and c odd, and that it is twice as fast as the multiplicative congruential method. The serial correlation coefficient between adjacent members of the sequence is satisfactory. By proper choice of the constants a and c, the correlation can be reduced still further.

806

The Generalization of Pseudo-Random Numbers on a Decimal Calculator, by J. Moshman (Oak Ridge Natl. Lab.); *J. Assoc. for Computing Mach.*, vol. 1, pp. 88–91; April, 1954.

A congruential method of generating pseudo-random numbers on a decimal machine is described. The conventional congruential methods are not satisfactory for decimal machines as they depend on finding the remainder after division by a power of two. Experimental and statistical tests of the methods are analyzed.

807

Addressing for Random-Access Storage, by W. W. Peterson (IBM Corp.); *IBM J. Res. and Dev.*, vol. 1, pp. 130–146; April, 1957.

Estimates of the searching time required for the location of a record in several types of storage systems, including the indexable method and sorting-file addressing, are derived. Detailed data and formulas for access-time for an "open" system of high flexibility and speed of access are provided. Experimental results for actual record files are quoted.

D-5: PROGRAMS—APPLICATIONS, ANALOG COMPUTERS

808

Computer Evaluation of High-Temperature Aircraft A-C Electrical System Designs, by W. E. Sollecito and D. A. Swann (G.E. Co.); *Applications and Industry*, no. 46 (*Trans. AIEE*, pt. II, vol. 78), pp. 434–444; January, 1960.

A design and analog computer study of a high-temperature aircraft ac electrical system is described. The study includes an extensive, nonlinear representation of the ac alternator and shows the effects of temperature, load, and regulator and exciter nonlinear behavior on system stability and transient performance. Torque-limiting techniques are evaluated and a computer approach for study of the reactive load division loop is developed.

809

Using an Electronic Simulator to Study the Compensation of Fluctuations in the Hot-Strip Thickness in a Rolling Mill, by S. A. Doganovskii and A. A. Fel'dbaum; *Automation Express*, vol. 1, pp. 22–23 (Excerpts); May 31, 1959 [Translation of: *Avtomatika i Telemekhanika*, vol. 20, pp. 192–205; February, 1959].

The theory, design, and application of an analog computer for compensating hot-strip thickness fluctuations in a rolling mill are discussed.

810

A Stack Effluent Radioisotope Monitor, by R. A. Harvey (G.E. Co.); IRE TRANS. ON NUCLEAR SCIENCE, vol. NS-6, pp. 20–28; December, 1959.

A system for simultaneously measuring and indicating each of several different radioisotopes in the stack effluent of an atomic energy facility is described. Analog computer techniques are employed for the solution of simultaneous equations and for the counting and scaling. Two types of presentations for the output are included.

811

The Use of an Analog Computer for Analysis of Control Mechanisms in the Circulation, by H. R. Warner (Latter Day Saints Hospital, Salt Lake City, Utah); PROC. IRE, vol. 47, pp. 1913–1916; November, 1959.

Two approaches to the study of regulation in the circulatory system are presented. One consists of programming, on an analog computer, equations to represent part of the system and then, using a suitable transducer, substituting the computer for the biological component. Simulation of a part of the mechanism which regulates arterial pressure (the carotid sinus) is discussed as an example. The other approach involves simultaneous solution of equations derived to represent each system component. Simulation of a transient disturbance in blood distribution (Valsalva maneuver) is presented to illustrate the use of this approach in predicting the role of each component in determining over-all system behavior.

812

Respiratory Control of Heart Rate: Laws Derived from Analog Computer Simulation, by M. Clynes (Rockland State Hospital, Orangeburg, N. Y.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 2–14; January, 1960.

The derivation, by means of analog computer simulation, of differential equations which relate the human heart rate to respiration is discussed. The simulation is made by the application of control-system theory. The only input to the computer is the respiration in the form of an electrical signal proportional to chest circumference. With this information, the computer calculates the timing of the heart from beat to beat. The close correspondence of the predicted and actual changes of heart rate for a wide variety of modes of breathing and for different individuals proves the validity of the equations.

E-1: MATHEMATICS—LOGIC—THEORETICAL MATHEMATICS

813

An Extension of Wiener Filter Theory to Partly Sampled Systems, by H. M. Robbins (Hughes Aircraft Co.); IRE TRANS. ON CIRCUIT THEORY, vol. CT-6, pp. 362–370; December, 1959.

The growing use of digital computers as components of control systems has given great importance to the study of linear systems which are partly sampled and partly continuous. The problem of optimizing the simplest possible mixed system consisting of an input filter with transfer function $K(s)$, a sampler with sampling interval T, and an output filter with transfer function $L(s)$, is considered. Given the power spectra of the input signal and the noise, the object is to find a realizable K and L which, in combination, minimize the mean-square difference between the output h and a "desired output" h_d. h_d is defined by a "desired transfer function" $G_d(s)$, not necessarily realizable, which would produce h_d from the input signal if the noise were absent. KL will, in general, contain factors periodic in s with period $2\pi j/T$, and such factors may be moved to either side of the sampler without changing the final output, thus introducing a considerable arbitrariness in K and L. However, since these periodic factors represent linear operations on discrete data (such as might be performed inside a digital computer), it is appropriate to separate them out. There are then three functions to be determined: the nonperiodic part of K, the nonperiodic part of L, and the remaining (periodic) factor of KL. Methods for determining these three functions are given. The interesting theoretical point is that the determination is not always unique. In general, there will be a finite number of distinct but equivalent solutions.

814

Serial Correlation in the Generation of Pseudo-Random Numbers, by R. R. Coveyou (Oak Ridge Natl. Lab.); *J. Assoc. for Computing Mach.*, vol. 7, pp. 72–74; January, 1960.

A convenient formula for the serial correlation coefficient in congruential pseudo-random generation processes is derived. Appropriate caveats governing its use are stated and further correlation coefficients to ensure the satisfactoriness of a given generating system are proposed.

815
Reduction of Runs in Multiparameter Computations, by H. Weil (University of Michigan); *J. Assoc. for Computing Mach.*, vol. 2, pp. 99–111; April, 1955.

Using statistical results based on the theory of the design of experiments, considerable economies effected in the evaluation of a multiparameter function are described. The statistical functions testing independence of neighboring points are developed and their use in numerical examples is considered.

E-2: MATHEMATICS—LOGIC—SYMBOLIC LOGIC, BOOLEAN ALGEBRA, NUMBER SYSTEMS

816
Toward Mechanical Mathematics, by H. Wang (IBM Corp.); *IBM J. Res. and Dev.*, vol. 4, pp. 2–22; January, 1960.

The results of a successful attempt at proving the approximately 400 theorems of *Principia Mathematica* which are strictly in the realm of logic, *viz.*, the restricted predicate calculus with equality, are reported. A number of other problems of the same type are also discussed. A new branch of applied logic, which may be called "inferential" analysis, and which treats proofs as numerical analysis does calculations, is suggested. It is possible that in the near future this new discipline can lead to machine proofs of difficult new theorems. An easier prepatory task is to use machines to formalize proofs of known theorems. This line of work may also lead to mathematical checks of new mathematical results comparable to the debugging of a program.

817
Solution to the Realizability Problem for Irredundant Boolean Branch-Networks, by L. Lofgren (Res. Inst. Natl. Defense, Stockholm); *J. Franklin Inst.*, vol. 268, pp. 352–377; November, 1959.

The existence problem for Boolean branch-networks (contact-networks), with only one branch for each different literal of the corresponding Boolean form, is a recognized key problem for the synthesis of combinatorial (nonoriented) switching circuits. It is equivalent to the topological problem of finding a necessary and sufficient condition for a matrix of integers mod 2 to be a circuit matrix (incidence matrix of a loop graph). This type of network, called irredundant, is defined for the 2- and n-terminal case, and a simple general solution to the existence and realizability problem is given. This solution is also a solution to the topological problem. The theory on nonoriented networks is readily extended to irredundant networks of oriented branches.

818
New Methods of Simplifying Boolean Functions, by R. L. Howard (Westinghouse Electric Corp.); *Applications and Industry*, no. 43 (*Trans. AIEE*, pt. II, vol. 78), pp. 134–143; July, 1959.

Two new methods for the simplification and transformation of Boolean functions are described and examples of each method are given. One method is used to convert the original function to a minimum sum of products and the other method is used to convert the original function to a minimum product of sums. Both methods utilize the same basic rules but they record the original function in different ways. A technique for shortening the basic procedure is also presented and illustrated by several examples. The procedure can be easily programmed for digital computer application.

819
Irredundant Disjunctive and Conjunctive Forms of a Boolean Function, by M. J. Ghazala (IBM Corp); *IBM J. Res. and Dev.*, vol. 1, pp. 171–176; April, 1957.

An algebraic method for the determination of the complete set of irredundant normal and conjunctive forms of a Boolean Function from a knowledge of its prime implicates is described. The method is readily programmable on a computer.

E-3: MATHEMATICS—LOGIC—NUMERICAL ANALYSIS

820
Curve Fitting with a Digital Computer, by C. W. Clenshaw (Natl. Physical Lab., Eng.); *Computer J.*, vol. 2, pp. 170–173; January 1960.

A modification of Forsythe's least-squares method of polynomial curve fitting is described. As in Forsythe, orthogonal polynomials are utilized to avoid ill-conditioning problems. By representing each polynomial within the computer by the coefficients in its Tchebycheff series, the method may be modified to save substantially on storage space.

821
A New Method of Computation of Square Roots Without Using Division, by D. Sarafyan (Vanderbilt University); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 23–24; November, 1959.

Newton's formula for computing $\sqrt[m]{n}=r$ is a special case of a more general formula $r_{i+1}=1/\alpha+\beta[\alpha n/r_i{}^{m-1}+\beta r_i]$. In the case of square roots, by selecting $\alpha+\beta=10^{p+1}$, it is possible with appropriate manipulations to avoid the division process in the Newton algorithm. Illustrative examples indicate how a convenient starting value may be selected.

822
On the Computation of Exponential and Hyperbolic Functions Using Continued Fractions, by N. Macon (Alabama Polytechnic Inst.); *J. Assoc. for Computing Mach.*, vol. 2, pp. 262–267; October, 1955.

Methods for computing e^x, tanh x, and, consequently, sinh x and cosh x using continued fractions are developed. The methods possess the characteristics of simplicity, rapid convergence over a wide range of x, and almost no generation of roundoff. Evaluation of these functions using continued fractions materially reduces computing time and generated errors compared with the use of power series truncated to give comparable accuracy.

823
Solution of Certain Large Sets of Equations on Pegasus Using Matrix Methods, by L. B. Wilson (Naval Construction Res. Est.); *Computer J.*, vol. 2, pp. 130–133; October, 1959.

Two methods for solving large sets of simultaneous equations whose matrix of coefficients is a continuant, or tridiagonal, supermatrix are described. The first method is an extension of the method of successive elimination and back substitution, and the second is similar to the Gauss-Seidel iteration method. The elimination method is more economical in storage space, is faster, and can handle more variables. The second method is appropriate for ill-conditioned sets of equations.

824
The Secant Method for Simultaneous Nonlinear Equations, by P. Wolfe (RAND Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 12–13; December, 1959.

A procedure for the simultaneous solution of a system of equations, not necessarily linear, is described. This procedure is a generalization of the secant or interpolation method for a function of one variable. From a set of trial solutions, a new set is calculated by matrix inversion using the pivotal method. Under certain conditions, convergence of an order higher than one but less than two can be demonstrated.

825
On the Increase of Convergence Rates of Relaxation Procedures for Elliptic Partial Difference Equations, by M. L. Juncosa and T. W. Mullikin (RAND Corp.); *J. Assoc. for Computing Mach.*, vol. 7, pp. 29–36; January, 1960.

In the numerical solution of elliptic partial differential equations, the rate of convergence of relaxation methods is sometimes affected by the relative proximity of certain points on the grid and boundary. It is shown that the removal of these points by Gaussian elimination does not affect the convergence of the Jacobi-Richardson and Gauss-Seidel iterations adversely. This indicates, though does not prove, that the elimination could improve the convergence rate for overrelaxation.

826
Boundary Contraction Solution of Laplace's Differential Equation II, by Tse-Sun Chow (Boeing Airplane Co.) and H. W. Milnes (General Motors Corp.); *J. Assoc. for Computing Mach.*, vol. 7, pp. 37–45; January, 1960.

The numerical solution of Laplace's equation for the circle is discussed, and the convergence of the solution obtained by the boundary contraction method to the analytic solution is considered. This convergence is shown to depend upon a relation between mesh sizes in the circumferential and radial directions. The error caused by boundary contraction can be made insignificant.

827
Stability of a Numerical Solution of Differential Equations—Part II, by W. E. Milne and R. R. Reynolds (Oregon State College);

J. Assoc. for Computing Mach., vol. 7, pp. 46–56; January, 1960.

The instability in Milne's method of solving differential equations numerically is shown to be avoidable by the occasional use of Newton's "three-eighths" quadrature formula in place of Simpson's rule. Part I (see Abstract No. 384) dealt with a single equation of first order. In Part II the analysis is extended to equations and systems of equations of higher order.

828
A Numerical Method for Solving Control Differential Equations on Digital Computers, by W. H. Anderson, R. B. Ball, and J. R. Voss (Bendix Aviation Corp.); *J. Assoc. for Computing Mach.*, vol. 7, pp. 61–68; January, 1960.

Frequently, as in missile-control systems, linear differential equations are simultaneous with nonlinear, but slower-acting, differential equations. The numerical solution of this type of system on a digital computer is significantly speeded up by approximating the forcing functions with polynomials, solving the linear equations exactly, and numerically integrating the nonlinear equations with Milne integration. Automatic interval adjustment is possible by comparing errors in the nonlinear integration. The interval selected is related to the shortest time constant of the nonlinear equations rather than the shortest of all the equations. With this system, both detailed transient response and steady-state conditions are revealed with a minimum of machine time.

829
Propagation of Truncation Errors in the Numerical Solution of Differential Equations by Repeated Closures, by H. J. Gray, Jr. (University of Pennsylvania); *J. Assoc. for Computing Mach.*, vol. 2, pp. 5–18; January, 1955.

A series expansion method that permits quantitative estimates of the truncation error arising in the course of the numerical solution of differential equations by the repeated closure method is described. It is complementary to the stability chart method, which indicates, only in a qualitative manner, any serious difficulties likely to arise in the numerical solution of a set of differential equations sufficiently linear to possess natural resonances.

830
Truncation Error in the Graeffe Root-Squaring Method, by G. P. Weeg (Michigan State University); *J. Assoc. for Computing Mach.*, vol. 7, pp. 69–71; January, 1960.

Estimates of the relative truncation error after p applications of the Graeffe root-squaring method for finding the zeros of a polynomial with real coefficients are derived. The relative error in the corresponding root of the original equation is only 2^{-p} as much as in the final equation.

831
A Generalization of a Theorem of Carr on Error Bounds for Runge-Kutta Procedures, by B. A. Galler and D. P. Rozenberg (University of Michigan); *J. Assoc. for Computing Mach.*, vol. 7, pp. 57–60; January, 1960.

A generalization of a theorem of Carr on error bounds for Runge-Kutta procedures is derived. The result covers a much wider class of Runge-Kutta procedures than Carr's theorem, and includes the Gill method commonly used on computers.

E-4: MATHEMATICS—LOGIC—THEORETICAL LINGUISTICS

832
On the Vibration of a Square Clamped Plate, by M. Abramowitz and W. F. Cahill (Natl. Bur. of Standards); *J. Assoc. for Computing Mach.*, vol. 2, pp. 162–168; July, 1955.

The calculation by the method of finite differences of the characteristic modes of vibration of a square plate clamped at all four edges is described. Methods of improving the numerical accuracy by decreasing the mesh, by improved approximations to the biharmonic operator ∇^2, and by using higher-order approximations for the boundary condition involving the normal derivative, are compared with each other, using as a norm the standard Rayleigh-Ritz-Weinstein method.

833
Toward a Model for Speech Recognition, by K. N. Stevens (M.I.T.); *J. Acoust. Soc. Am.*, vol. 32, pp. 47–55; January, 1960.

An approach to the design of a machine for the recognition and synthesis of speech is proposed, with particular emphasis on problems of acoustical analysis. As a recognizer, the proposed machine accepts a speech wave at its input and generates a sequence of phonetic symbols at its output; as a synthesizer, it accepts a sequence of symbols at its input and generates a speech wave. Coupling between the acoustical speech signal and the machine is achieved through two peripheral units: one an analog filter set or equivalent, and the other a model of the vocal tract. Between the analog filters and the phonetic output, the signal undergoes an intermediate form of representation that is related to vocal-tract configurations and excitations, but is not necessarily described specifically in these terms. Each stage of analysis is performed by synthesis of a number of alternative signals or patterns according to rules stored within the machine and by comparison of the synthesized patterns with the input signals that are under analysis. Possible advantages of the proposed method of analysis are discussed and an experimental study based on the general analysis approach is described In this study, a method for the determination of the frequencies of vocal-tract resonances from the speech wave is simulated on a digital computer.

E-5: MATHEMATICS—LOGIC—INFORMATION THEORY

834
A Class of Systematic Codes for Non-Independent Errors, by N. M. Abramson (Stanford University); IRE Trans. on Information Theory, vol. IT-5, pp. 150–157; December, 1959.

A class of systematic codes which will correct all single errors and all double errors which occur in adjacent digits is discussed. These codes use significantly fewer checking digits than codes which correct all double errors. In addition, because of inherent regularities in their structure, these codes may be instrumented in a strikingly simple fashion.

835
On Upper Bounds for Error Detecting and Error Correcting Codes of Finite Length, by N. Wax (University of Illinois); IRE Trans. on Information Theory, vol. IT-5, pp. 168–174; December, 1959.

Upper bounds for error-detecting and error-correcting codes are obtained. One upper bound is found by exploiting the geometrical model of coding introduced by Hamming. The volume of an appropriate geometrical body is compared with the volume of the unit cube, in getting the first upper bound. An improvement on this upper bound can be found by introducing a mass density function, and comparing the mass of the body with the mass of the unit cube. A comparison is made with known upper bounds, and with best codes found thus far. The improved upper bound given here is frequently somewhat smaller than previously known upper bounds.

836
A New Group of Codes for the Correction of Dependent Errors in Data Transmission, by C. M. Melas (IBM Corp.); *IBM J. Res. and Dev.*, vol. 4, pp. 58–65; January, 1960.

Any configuration of multiple related errors is shown to be correctable by a class of codes using two groups of parity bits, one defining the error pattern and the other determining the location of errors within a block. In particular, the correcting of error bursts with minimal redundancy is achieved. The codes are implemented relatively simply by using maximum-length shift-register sequences.

837
Design Methods for Maximum Minimum-Distance Error-Correcting Codes, by J. E. MacDonald (IBM Corp.); *IBM J. Res. and Dev.*, vol. 4, pp. 43–58; January, 1960.

An upper-bound on the minimum distance between code points of an error-correcting code which depends on g, the number of code points, and n, the number of binary symbols per code point, is derived. The bound is complementary to the classic Hamming bound. Construction methods for codes which achieve the minimum distance are presented. Sixteen code types, three for g odd, six for g even and seven for $g=2^K$, are classified.

E-6: MATHEMATICS—LOGIC—LINEAR PROGRAMMING

838
Sequential Machines, Ambiguity, and Dynamic Programming, by R. Bellman (RAND Corp.); *J. Assoc. for Computing Mach.*, vol. 7, pp. 24–28; January, 1960.

The problem of determining testing procedures, which will transform a sequential machine to a known state from an initial state in which only the set of possible states

is given, is discussed. The concept of ambiguity is introduced into the problem and the functional equation approach of dynamic programming is applied.

839
An Introduction to Linear Programming with Applications to Data Reduction Problems, by F. Reed (Naval Ordnance Test Station); *U. S. Govt. Res. Repts.*, vol. 33, p. 76(A); January 15, 1960. PB 143 341 (Order from LC mi $3.30, ph $7.80).

An introduction to the mathematical theory of linear programming is presented. The material covered is sufficient to enable the applied mathematician to solve a great number of practical problems. As illustrations of the usefulness of the technique, two problems related to the field of data reduction are solved.

F: PERSONNEL

840
Symposium on the Selection and Training of Programmers—I: A Business Users Approach, by H. W. Gearing (Metal Box Co. Ltd.); *Computer J.*, vol. 2, pp. 107–109; October, 1959.

The main aspects of recruiting and training programmers for commercial programs are outlined. A programming team should have both academically and business-orientated representation. Clear procedures, such as flowcharting, that make a finished program intelligible to many users must be adhered to. A systematic and methodical approach to problems is essential to a programmer of business problems.

J: SUMMARIES AND REVIEWS

841
Impact of Computer Developments, by S. M. Humphrey (Booz, Allen and Hamilton); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 16–19; December, 1959.

The trend of computer developments is surveyed. A need for larger memories and the importance of careful planning to minimize the impact of obsolescence are noted. A universal computer-oriented credit-card system for all financial transactions is predicted, provided that human factors, such as concern for information security and resistance to habit change, can be overcome.

842
The State of the Art—a) Commercial Computers in Britain, June 1959, by J. A. Goldsmith (Robson, Morrow and Co., Ltd.); *Computer J.*, vol. 2, pp. 97–99; October, 1959.

The development of commercial computer utilization in Britain is described. At present, about 100 installations are operating and the days of experimenting and loose predictions are over. Computers are generally recognized as useful tools in many applications. Shortage of properly-trained programmers and the paucity of hire-time facilities are two features particularly noted.

843
The State of the Art—b) Computers in British Universities, by A. S. Douglas (University of Leeds); *Computer J.*, vol. 2, pp. 100–102; October, 1959.

The status of computing in British Universities is discussed. The trend is for several universities to share a computing installation, in view of the high cost of computers, or for computer developments to take place in government or industry-supported pseudo-universities. Considerable service work is accepted to contribute towards upkeep costs. At present, fewer than ten computers are available for university use.

844
Development of Japanese Digital Computers, by S. Takahashi (Electrotechnical Lab., Tokyo); *Computer J.*, vol. 2, pp. 122–129; October, 1959.

The Japanese computer industry is surveyed and the currently-employed transistor and parametron technologies, including a parametron-core matrix memory, are briefly described. A complete tabulation of individual Japanese machines is provided.

SUBJECT INDEX

AUTHOR INDEX

PGEC Notices

To All PGEC Chapter Officers

The PGEC *News* page of these TRANSACTIONS is open for your announcements and report of activities. Deadline is the first of the month, two months ahead of the date of issue. Send all items to the *Editor*.

A NOTE FROM THE CHAIRMAN

TO MEMBERS OF PGEC:

Since the last Chairman's report, there have been two meetings of the PGEC Administrative Committee. The first was the annual meeting at IRE Headquarters in March, and the second was at the WJCC in San Francisco in May. Also, the NJCC has held a particularly important meeting. In this, my first communication to you, I will touch upon several highlight topics of these recent meetings without going into chronological detail.

NEW ADMINISTRATIVE COMMITTEE MEMBERS AND OFFICERS

The new PGEC officers and Administrative Committee members, elected and approved at the annual meeting, took office April 1. It is my pleasure to welcome the six members of the Administrative Committee who are beginning their three-year period of service. They are well known in the computer world and in the IRE. We look forward to their valuable counsel and active participation in guiding the affairs of PGEC. They are:

William E. Bradley	Samuel Lubkin
William T. Clary	Roger L. Sisson
Louis Fein	Luverne J. Spieker

Our new Vice-Chairman is Frank E. Heart, who has been active in the Administrative Committee and who was General Chairman of the highly successful 1959 EJCC at Boston. Our Secretary-Treasurer for 1960–1961 is Edward D. Zimmer, who has been active in the Twin Cities (Minnesota) PGEC Chapter.

PROPOSED IFIPS

The official relationship of IRE and PGEC to the newly formed International Federation of Information Processing Societies (IFIPS) has been discussed previously in these columns, as has the need for an enlarged scope U. S. "society of societies" to replace the present NJCC. Such a federation is required as a mechanism for providing unified U. S. representation to IFIPS, and for performing other appropriate functions with proper authority.

At the May 4 NJCC meeting in San Francisco, a preliminary draft of the constitution for a proposed American Federation of Information Processing Societies (AFIPS) was introduced. The full committee then undertook a careful point-by-point review, leading to a revised draft which has subsequently been submitted to the administrative bodies of the three "founder societies" for review. NJCC's goal is to accomplish ratification by the founder societies before the end of the year.

The constitution of the proposed federation is similar in many respects to that of the American Institute of Physics, which served as an initial pattern. It would be premature to go into detail at this time, but it may be enlightening to quote from the Constitution's Statement of Purposes: ". . . to serve the public by making available . . . reliable communications as to information processing and its progress; to cooperate with local, national, and international organizations or agencies on matters pertaining to information processing; to serve as representative of the United States of America in international organizations with like interests; to promote unity and effectiveness of effort among all those who are devoting themselves to information processing by research, by application of its principles, by teaching, or by study; and to foster the relations of the sciences of information processing to other sciences and to the arts and industries. In pursuing these purposes, the Federation shall do nothing that is in direct competition with activities of its member societies."

BIENNIAL PGEC MEMBERSHIP SURVEY

The PGEC membership surveys of 1956 and 1958 have made available extremely valuable information pertaining to the electronic computer profession. At its May 3rd meeting, the Administrative Committee authorized a similar survey for 1960. Keith Uncapher was named chairman of the committee to conduct the survey. At this writing, the survey questionnaire and letter of explanation are being debugged, based on experience of the two previous runs. Your questionnaire will be mailed to you about September 1, for return before October 1. The usual meticulous precautions will be observed to insure confidential treatment of individual data. The report will be published early in 1961. Your diligence in filling out and returning your questionnaire will contribute to the accuracy of the only definitive survey concerned exclusively with our field.

MEMBERSHIP REPORT

At the March annual meeting, Membership Chairman Robert Blakely reported the following membership as of December 31, 1959: full members, 8129; student members, 679; affiliates, 66; total, 8874. This represents an increase of 1314 over the previous year. PGEC continues to be the largest professional group in IRE.

On the basis of a favorable recommendation by the PGEC Membership Committee, the Administrative Committee voted at its May meeting to accredit the Instrument Society of America for Affiliate memberships. This action brings to fifteen the number of societies whose members may apply for Affiliate membership in PGEC. These societies are listed on the back cover of this issue.

ARNOLD A. COHEN
Chairman

NOTICE OF ADMINISTRATIVE COMMITTEE MEETING AT WESCON

The next meeting of the PGEC Administrative Committee will be at some time during WESCON in Los Angeles, August 23–26, 1960. Any PGEC member who wants to attend the Administrative Committee meeting should so inform the Secretary-Treasurer of PGEC, Edward D. Zimmer, at Control Data Corp., 501 Park Ave., Minneapolis 15, Minn. Mr. Zimmer can then provide the data as to time and place of the meeting.

PGEC REPRESENTATIVES TO NJCC

Frank E. Heart and Harry Goode are the new PGEC representatives to NJCC (The National Joint Computer Committee). They join W. Buchholz and W. H. Ware in in this assignment. PGEC Chairman Arnold Cohen is the fifth representative, ex officio. Send any thoughts you wish to express on the subject of NJCC to any one of your representatives.

PGEC MEMBERSHIP SURVEY

The biennial PGEC membership survey is now getting under way and you shortly will receive a questionnaire designed to extract from you (as painlessly as possible) information that will give all of us a good view of the economic and job-distribution statistics of our profession. Please cooperate by returning your questionnaire promptly.

Notices

This *Notices* Section is open to all who have an announcement of a conference, symposium, session, publication, or other artifact of interest to the PGEC membership. Please send announcements to the *Editor*, who will put them in the first available issue. The right is reserved to edit the announcements, and to decide whether they indeed are aimed at our audience.

COMING MEETINGS

D.R.I. SEVENTH ANNUAL SYMPOSIUM ON COMPUTERS AND DATA PROCESSING

Denver Research Institute of the University of Denver is sponsoring its Seventh Annual Symposium on Computers and Data Processing on July 28–29, 1960 at the Stanley Hotel, Estes Park, Colorado. Three sessions are planned on components, logic design, and the philosophy of computer design. Write to W. H. Eichelberger, Arrangements Chairman, Denver Research Institute, Denver 10, Colo., for all details.

ANNUAL ACM MEETING

The fifteenth Annual Meeting of the Association for Computing Machinery will be held at Marquette University, Milwaukee, Wis., on August 23–25, 1960. Local arrangements will be under the direction of Prof. Arthur Moeller of Marquette. Contributed and invited papers on all phases of analog and digital computation, business applications, and data processing will be presented. Round-table discussions and a "Hall of Discussions" are planned.

WESCON

August 23–26, at the Ambassador Hotel and Pan-Pacific Auditorium in Los Angeles. (*See PGEC News.*)

INTERNATIONAL SYMPOSIUM ON DATA TRANSMISSION

The Benelux Section of IRE is sponsoring an International Symposium on Data Transmission in Delft, The Netherlands, on September 19–20, 1960. The symposium will be concerned with the problems of transmitting and receiving information in digital form. Particular emphasis will be placed on the behavior of practical communication networks, including existing telephone systems, existing and planned military systems, and schemes of the future, such as those that use satellites. The aim of the symposium will be to reduce the gap now existing between theory and practice. The symposium will be conducted in English. The papers scheduled give a good representation of work being carried on in the U.S. as well as in Europe. For further information, contact B. B. Barrow, Secretary, IRE Benelux Section, Postbus 14, Den Haag, The Netherlands.

AIEE SYMPOSIUM ON SWITCHING CIRCUIT THEORY AND LOGICAL DESIGN

The symposium will consist of sessions held during the AIEE Fall General Meeting in Chicago, October 9–14, 1960. For further information, write to Thomas H. Mott, Jr., RCA Laboratories, Princeton, N. J.

POWER INDUSTRY COMPUTER APPLICATIONS CONFERENCE

The Second Power Industry Computer Applications Conference sponsored by AIEE will be held in St. Louis, November 9–11, 1960. Complete details are available from E. L. Harder, Analytical Department 4113, Westinghouse Electric Corp., East Pittsburgh, Pa.

EJCC

New York, N. Y., December, 1960.

CALL FOR PAPERS

COMPUTER ISSUE OF PROCEEDINGS OF THE IRE

The Institute of Radio Engineers will publish a special Computer Issue of PROCEEDINGS in January, 1961.

You are invited to submit a paper for publication in this issue. Papers on *new developments* in digital and analog computers are desired, especially in the following areas:

- Components
- Storage devices and systems
- Arithmetic, logic, and control elements, including self-organizing machines
- Input and output elements, including voice and pattern recognition, and displays
- Logical design, including use of computers in logical design
- Circuit design, including use of computers in circuit design
- General-purpose computers and data processors
- Computers optimized for special roles
- Large systems in which computers play a role
- Digital communication
- Microminiaturization.

The deadline for papers is July 29, 1960. Papers must include a 100-word abstract. Three copies of the paper and illustrations are required. Also include one set of illustrations that are reproducible, and photographs and biographies of the authors. They should be sent to E. K. Gannett, Managing Editor, 1 East 79th Street, New York 21, N. Y. Clearly indicate that the paper is intended for the Computer Issue.

Computer Issue Editorial Board: H. T. Larson, L. C. Hobbs, K. W. Uncapher.

1960 EASTERN JOINT COMPUTER CONFERENCE

The 1960 Eastern Joint Computer Conference will be held in December 13–15 at the Hotel New Yorker and the Manhattan Center in New York City. This tenth Eastern Joint meeting will stress papers describing significant and interesting accomplishments in the field. Contributed papers may pertain to any aspect of computer application or development, but should report achievement.

Anyone wishing to submit a paper for consideration by the Program Committee is requested to forward four copies of an abstract as well as four copies of the proposed paper. The summary should consist of approximately 1000 words and will be used by the Program Committee to evaluate and select papers.

All copies of abstracts and summaries must be received by the Chairman of the Program Committee on or before August 13, 1960.

Elmer C. Kubie, Chairman
Program Committee 1960 EJCC
Computer Usage Company, Inc.
18 East 41st Street
New York 17, New York

In an attempt to build a program of the highest possible quality, no parallel sessions are planned and a $300 prize will be awarded for the best presentation of a paper at the Conference.

HFE SPECIAL ISSUE

The IRE TRANSACTIONS ON HUMAN FACTORS IN ELECTRONICS is planning to devote its March, 1961, issue to the topic, "Automation of Human Functions," and manuscripts for that issue are hereby solicited.

"Automation of Human Functions" is to be understood as including all techniques whereby the load on the human operator of electronic systems may be lessened through increased automation of his task. Automatic pattern recognition, problem solving, and decision making are examples of subjects included in this topic. Papers need not make original contributions but may review material already published.

To be considered, manuscripts must be received by October 15, 1960, and must comply with IRE standards. Manuscripts should be mailed to the guest editor:

Dr. Thomas Marrill
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge 38, Mass.

SERVICE AVAILABLE

Dr. Geoffrey Knight, Jr., wholse organization, Cambridge Communications, prepares the Abstracts printed in IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, has now made these abstracts available on file cards as well. Write him at 238 Main Street, Cambridge 42, Mass., for further information.

INFORMATION FOR AUTHORS

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS is published quarterly, in March, June, September, and December, with a distribution of over 9000 copies, largely to engineers, logicians, and supervisors in the computer field. Its scope includes the design, theory, and practice of electronic computers and data-processing machines, digital and analog, and parts of certain related disciplines such as switching theory and pulse circuits.

If a paper of widespread interest beyond the computer field is submitted, it will be recommended to the Editor of PROCEEDINGS OF THE IRE for publication. If our reviewers feel that a paper should be submitted to a different IRE TRANSACTIONS, we will so recommend to the author.

Publication time in IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, from receipt of the original manuscript to mailing of the issue, is normally in excess of 5 months, but can be made as little as $3\frac{1}{2}$ months if the occasion demands and the manuscript is carefully prepared.

To avoid delay, please be guided by the following suggestions:

A. Process for Submission of a Technical Paper

1) Send to the Editor three copies of your manuscript, each copy complete with illustrations. (For Letters to the Editor, two copies will do.)

2) Enclose originals for the illustrations, in the style described below. Alternatively, be ready to send the originals immediately upon acceptance of the paper.

3) Enclose a separate sheet giving your preferred address for correspondence and return of proofs.

4) Enclose a technical biography and photograph of each author, or be ready to supply these upon acceptance of the paper. For biography style, see any IRE journal.

5) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit that have appeared or will appear before other audiences.

B. Style for Manuscript

1) Typewrite, double or $1\frac{1}{2}$ space; use one side of sheet only. (Good office-duplicated copies are acceptable.)

2) Provide an informative 100- to 250-word summary (abstract) at the head of the manuscript. It will appear with the paper and also separately in PROCEEDINGS OF THE IRE.

3) Provide a separate double-spaced sheet listing all footnotes, beginning with "*Received by the PGEC _____," and "†(Affiliation of author)," and continuing with numbered references. Acknowledgment of financial support is often placed at the end of the asterisk footnote.

4) Give complete references, insofar as possible. See footnotes in this or previous issues for examples of IRE style. References will be printed as footnotes in the column where first mentioned.

5) You may choose to provide a "Bibliography" at the end of the paper, with items referred to by a numeral in square brackets, *e.g.*, [12], to supplement or supplant footnote references.

6) Provide a separate sheet listing all figure captions, in proper style for the typesetter, *e.g.*: "Fig. 1—Example of a disjoint and distraught manifold."

C. Style for Illustrations

1) Originals for illustrations should be sharp, noise-free, and of good contrast. We regret that we cannot provide drafting or art service.

2) Line drawings should be in India ink on drafting cloth, paper, or board. Use $8\frac{1}{2} \times 11$ inch size sheets if possible, to simplify handling of the manuscript.

3) On graphs, show only the coordinate axes, or at most the major grid lines, to avoid a dense, hard-to-read result.

4) All lettering should be large enough to permit legible reduction of the figure to column width, perhaps as much as 4:1.

5) Photographs should be glossy prints, of good contrast and gradation, and any reasonable size.

6) Number each original on the back, or at the bottom of the front.

7) Note item B-6 above. Captions lettered on figures will be blocked out in reproduction, in favor of typeset captions.

Mail all manuscripts to:

Dr. Howard E. Tompkins, *Editor*
IRE TRANSACTIONS ON ELECTRONIC COMPUTERS
Electrical Engineering Department
University of New Mexico
Albuquerque, N. Mex.

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

Volume EC-9 June, 1960 Number 2

REVIEWS OF BOOKS AND PAPERS IN THE COMPUTER FIELD

(See front cover for main contents of issue)

Affiliate Status

Members of the professional societies listed below, who are not IRE members, may become AFFILIATES of the PGEC (Professional Group on Electronic Computers), and thus receive IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, by payment of $8.50 annually. Apply to IRE Headquarters, 1 East 79th St., New York 21, N. Y.

(IRE members who join PGEC are currently assessed $4.00 per year.)

Professional Societies Approved for Affiliates to PGEC

American Institute of Electrical Engineers
American Management Society
American Mathematical Society
American Physical Society
American Society of Mechanical Engineers
Association for Computing Machinery
Institute of the Aeronautical Sciences
Institution of Electrical Engineers (London)
Instrument Society of America
Mathematical Association of America
National Association of Cost Accountants
National Machine Accountants Association
Operations Research Society of America
Society for Industrial and Applied Mathematics
Society of Automotive Engineers